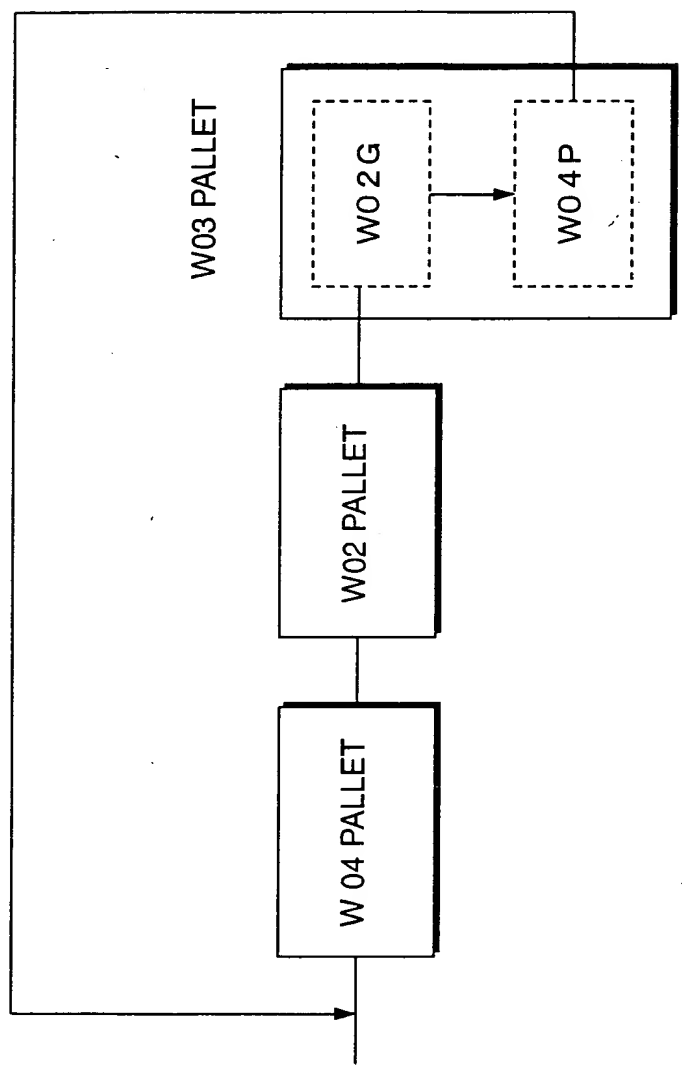
09/868893

Fig. 1



09/868893

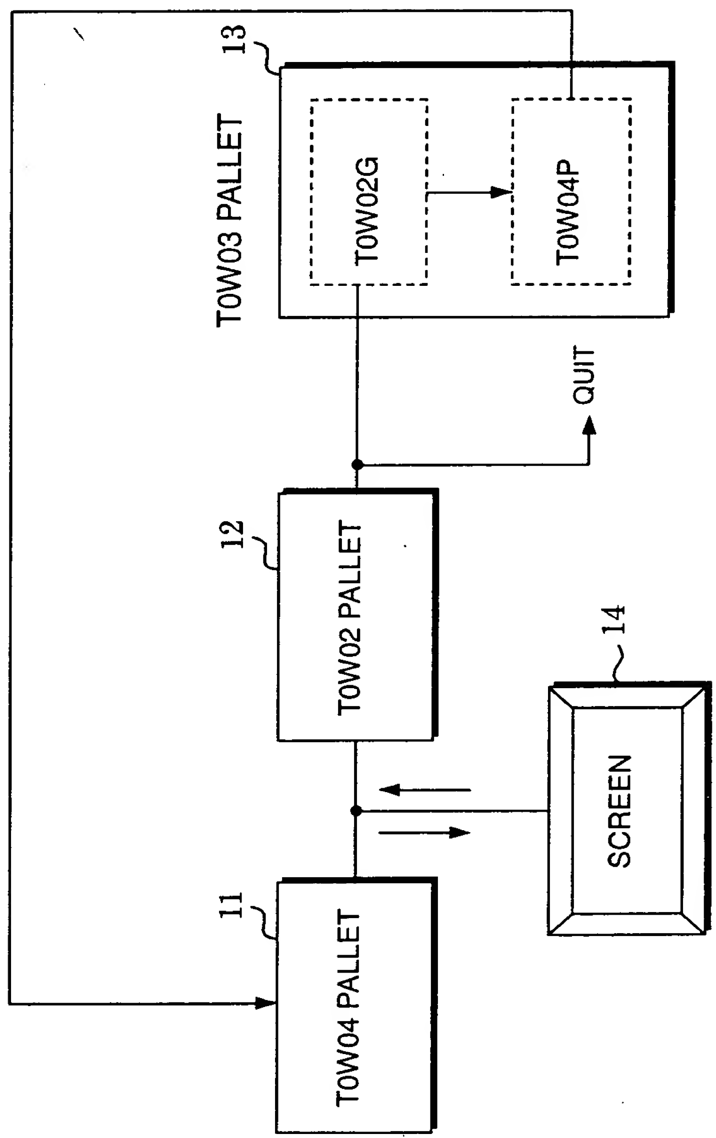


Fig. 2

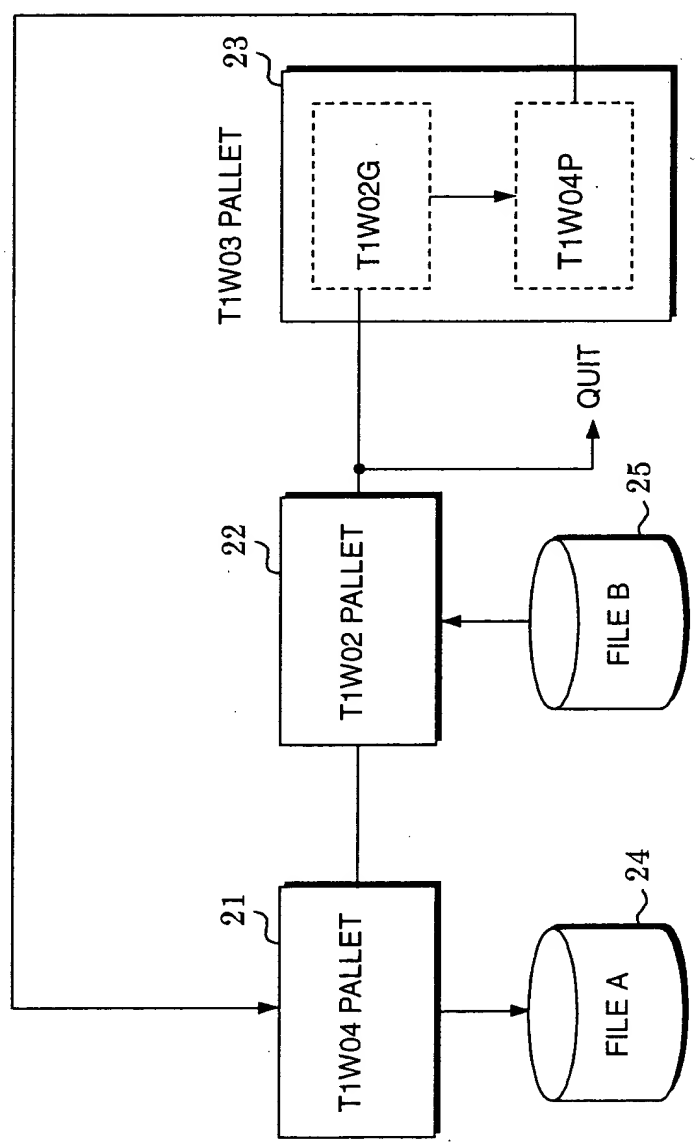

T1W03 PALLET
T1W02G
T1W04P
23
QUIT
22
T1W02 PALLET
25
FILE B
21
T1W04 PALLET
24
FILE A

Fig. 3

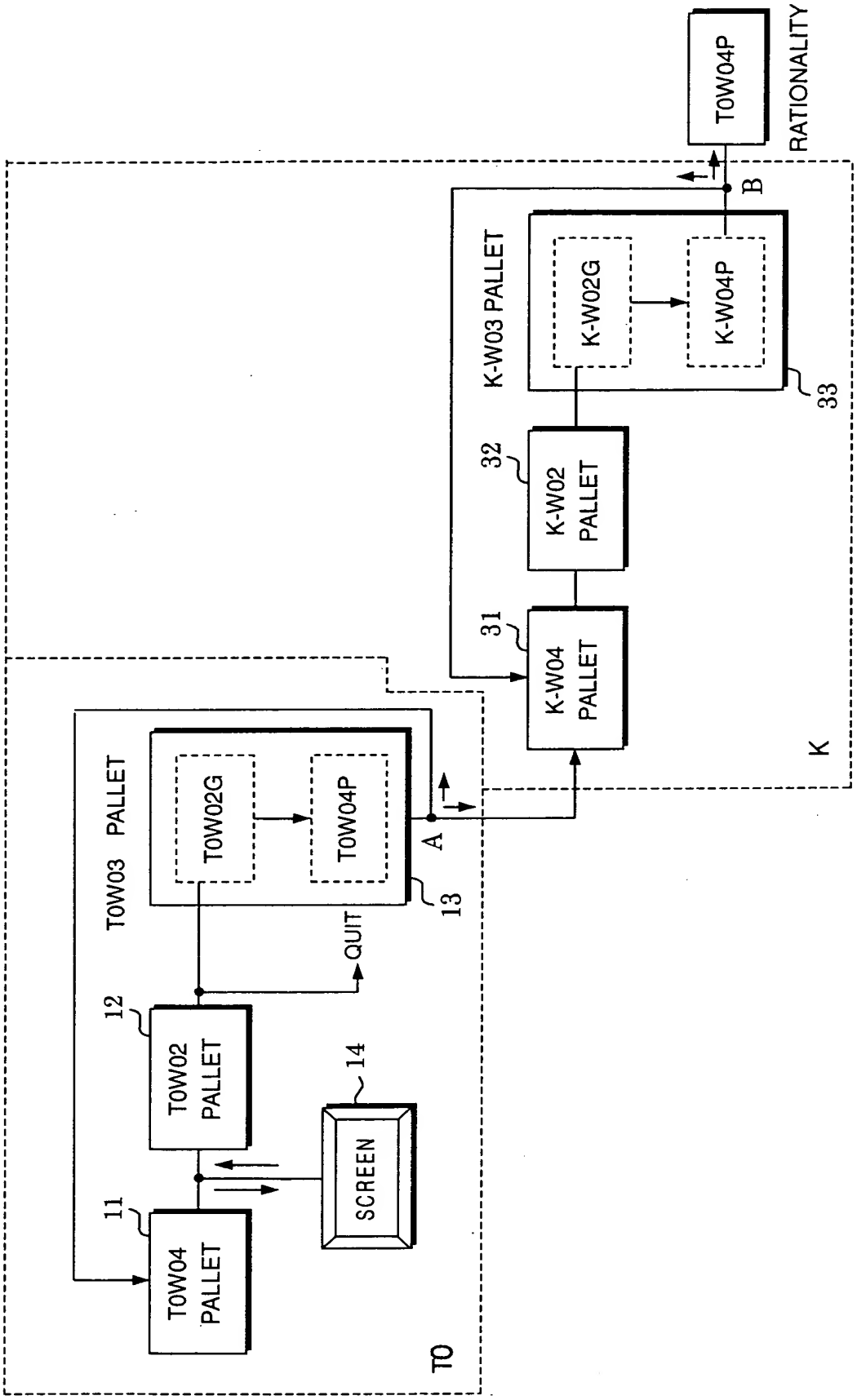
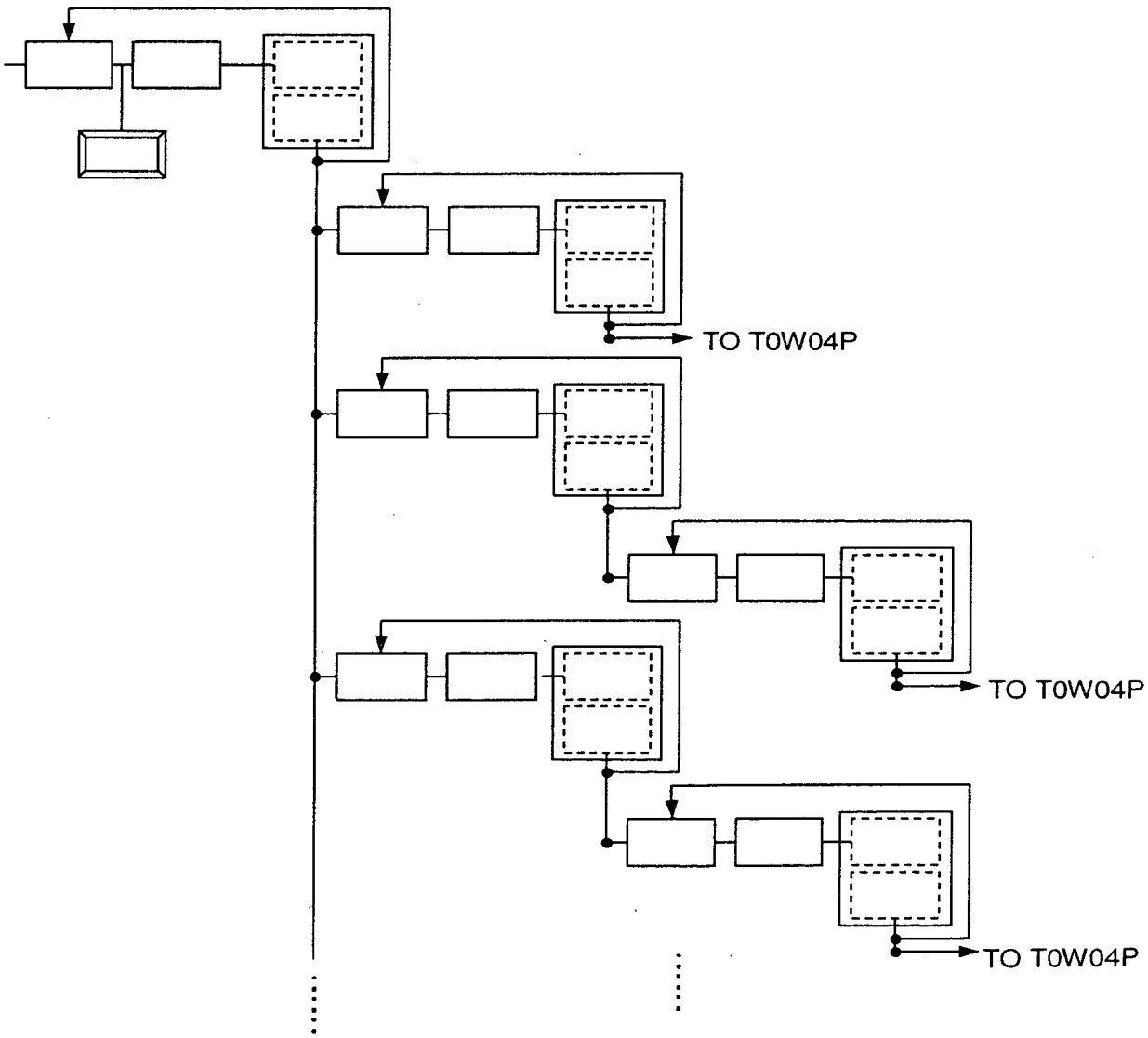

RATIONALITY
T0W04P
B
K-W03 PALLET
K-W02G
K-W04P
33
32
K-W02 PALLET
31
K-W04 PALLET
K
A
T0W03 PALLET
T0W02G
T0W04P
13
QUIT
12
T0W02 PALLET
14
SCREEN
11
T0W04 PALLET
T0

Fig. 4

Fig. 5



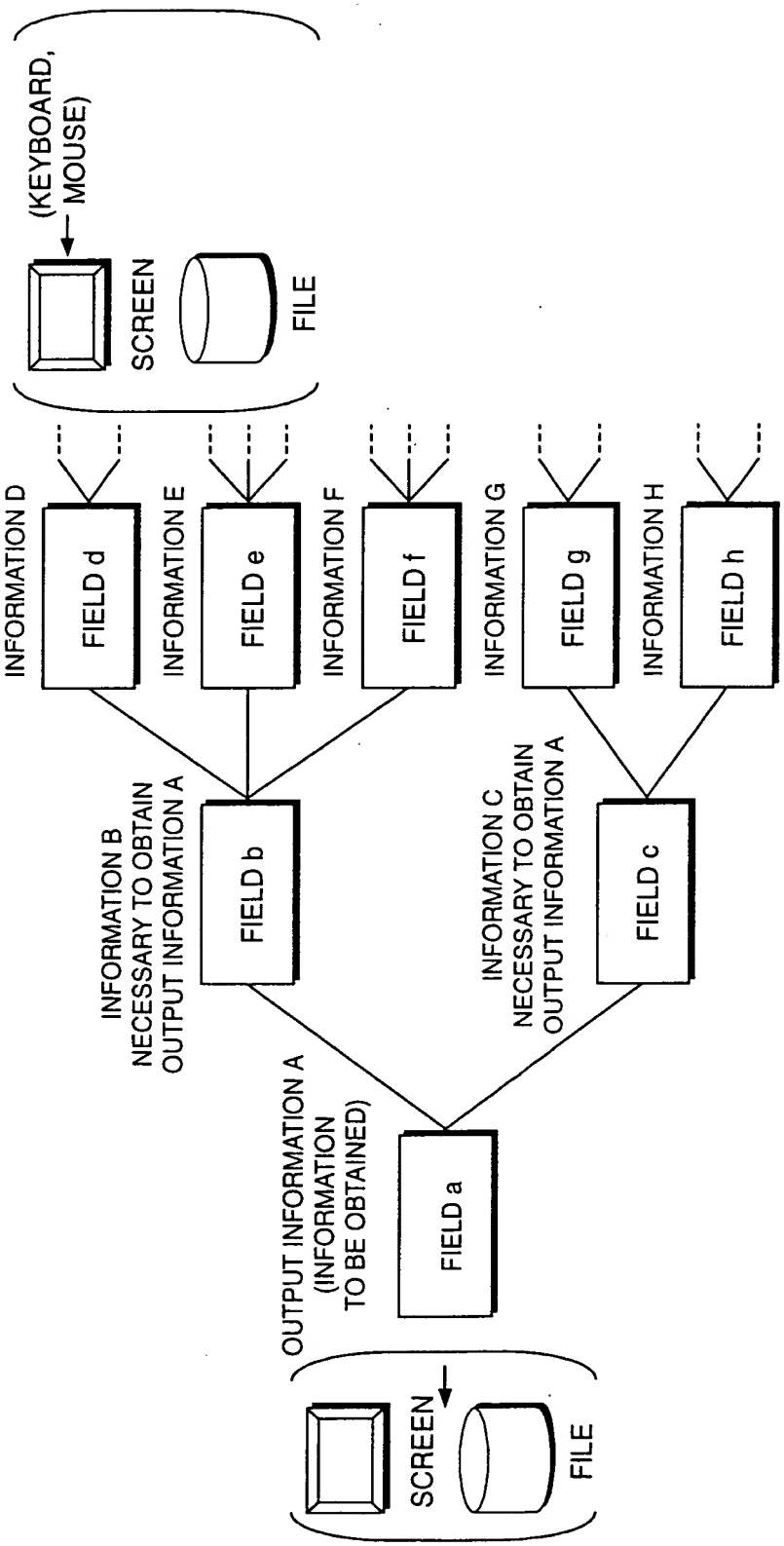

(KEYBOARD, MOUSE)
SCREEN
FILE
INFORMATION D
FIELD d
INFORMATION E
FIELD e
INFORMATION F
FIELD f
INFORMATION G
FIELD g
INFORMATION H
FIELD h
INFORMATION B NECESSARY TO OBTAIN OUTPUT INFORMATION A
FIELD b
INFORMATION C NECESSARY TO OBTAIN OUTPUT INFORMATION A
FIELD c
OUTPUT INFORMATION A (INFORMATION TO BE OBTAINED)
FIELD a
SCREEN
FILE

Fig. 6

Fig. 7

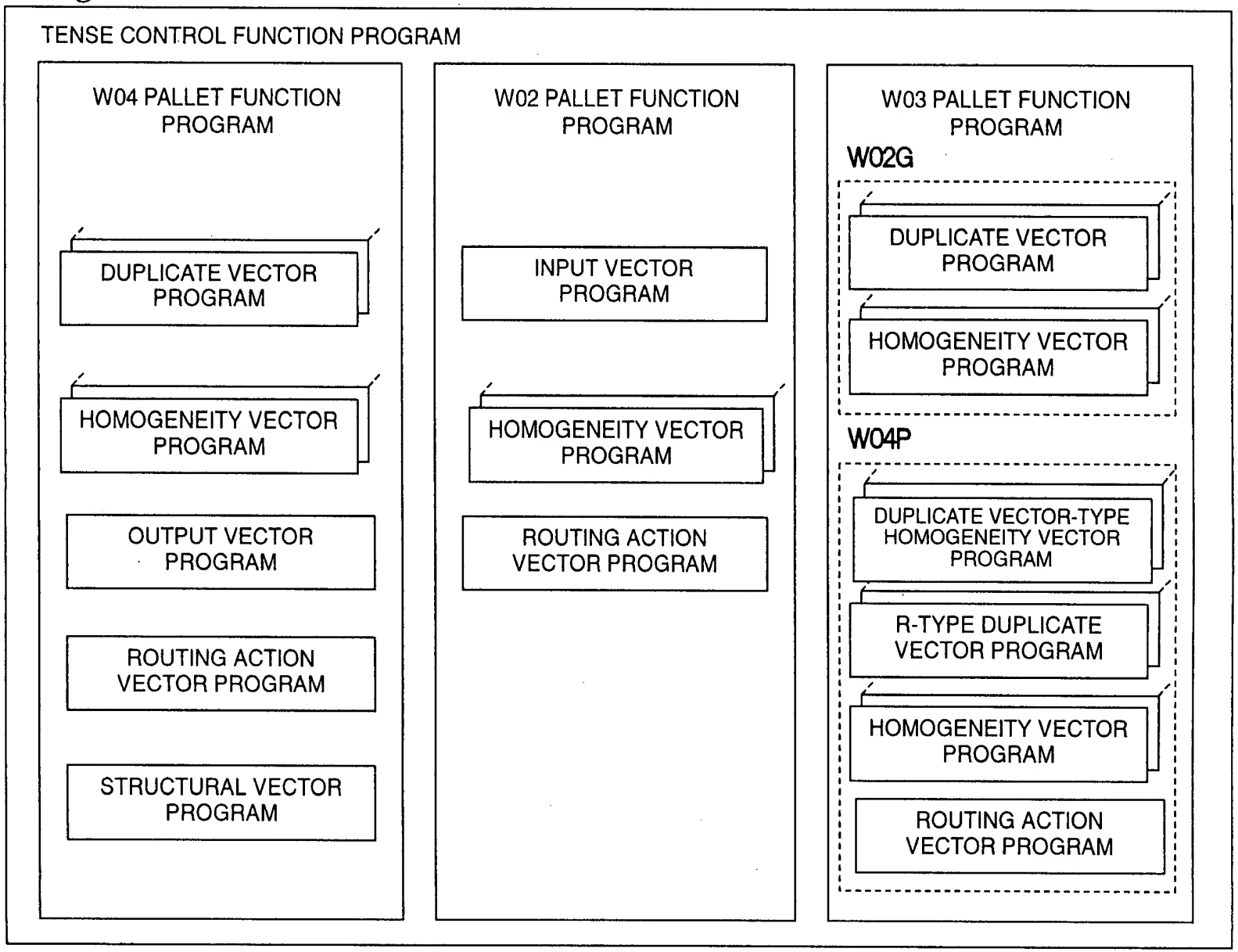


Fig. 8

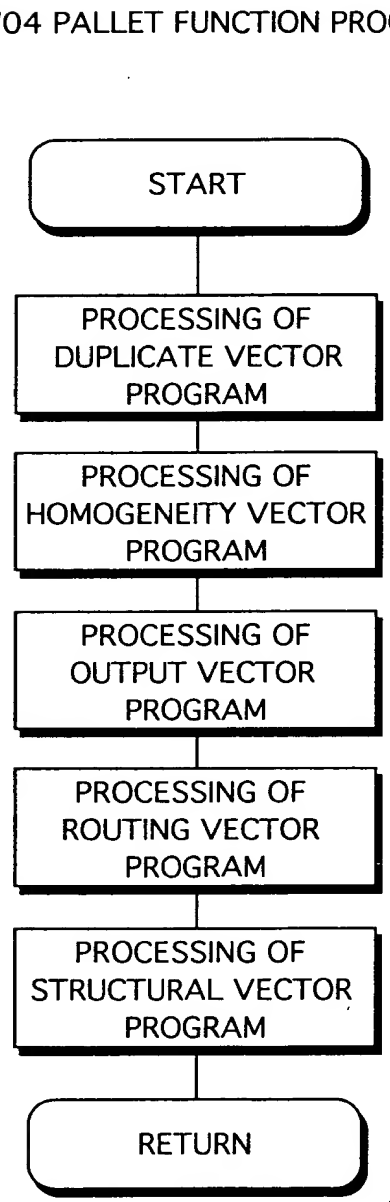

W04 PALLET FUNCTION PROGRAM
START
PROCESSING OF DUPLICATE VECTOR PROGRAM
PROCESSING OF HOMOGENEITY VECTOR PROGRAM
PROCESSING OF OUTPUT VECTOR PROGRAM
PROCESSING OF ROUTING VECTOR PROGRAM
PROCESSING OF STRUCTURAL VECTOR PROGRAM
RETURN

Fig. 9

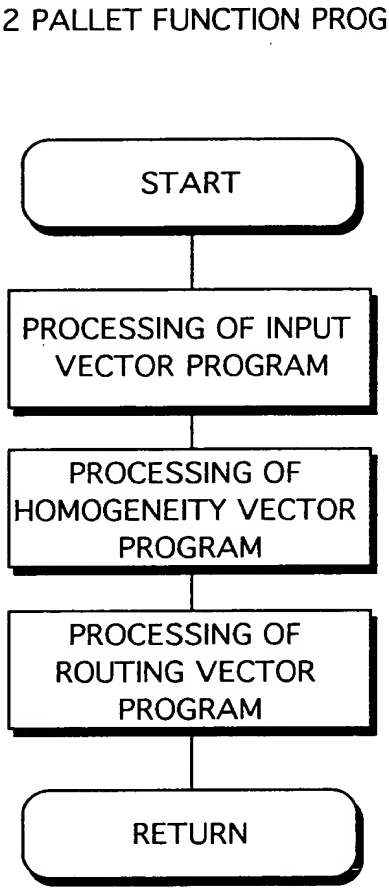

W02 PALLET FUNCTION PROGRAM
START
PROCESSING OF INPUT VECTOR PROGRAM
PROCESSING OF HOMOGENEITY VECTOR PROGRAM
PROCESSING OF ROUTING VECTOR PROGRAM
RETURN

Fig. 10

W03 PALLET FUNCTION PROGRAM

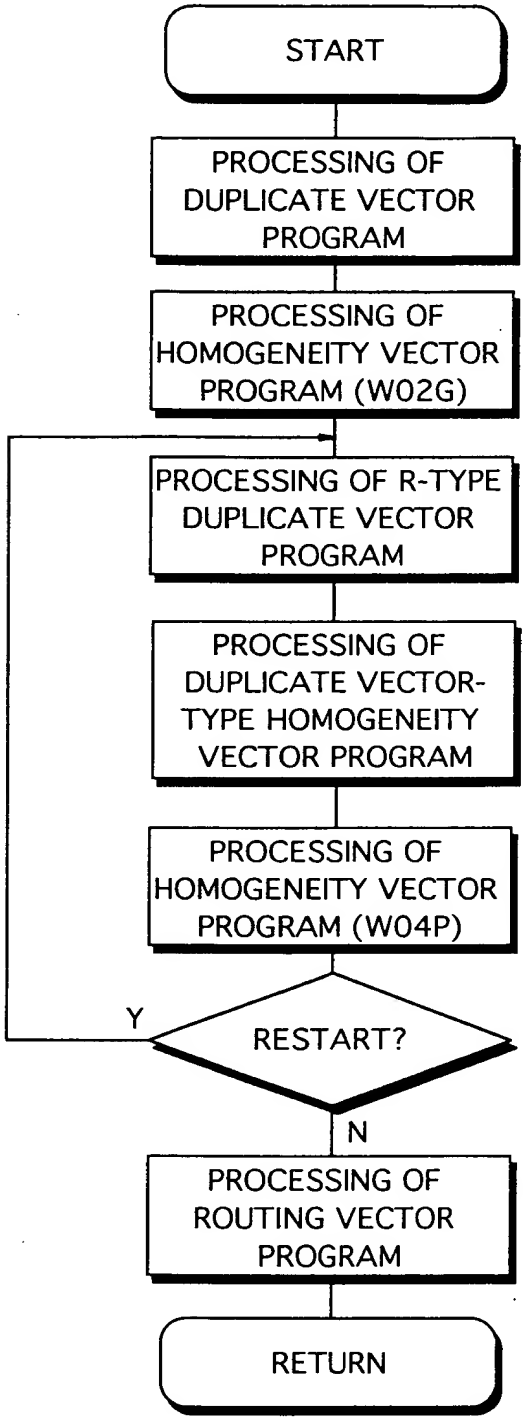


Fig. 11

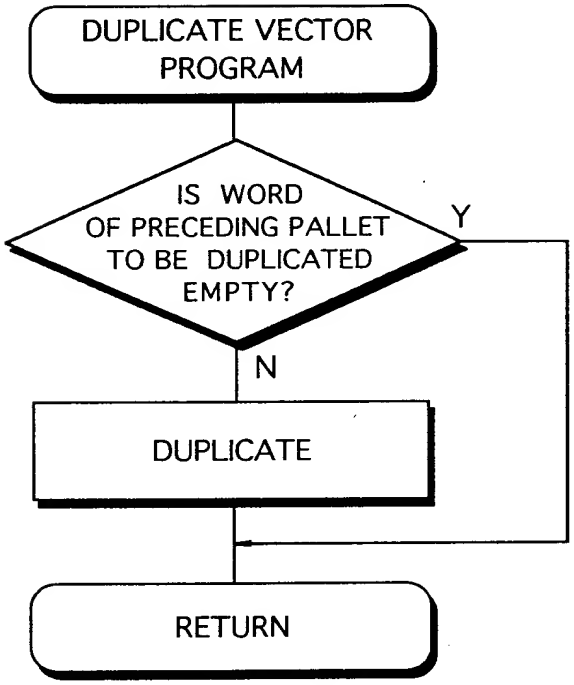

DUPLICATE VECTOR PROGRAM
IS WORD OF PRECEDING PALLET TO BE DUPLICATED EMPTY?
Y
N
DUPLICATE
RETURN

Fig. 12

HOMOGENEITY VECTOR PROGRAM (W04)

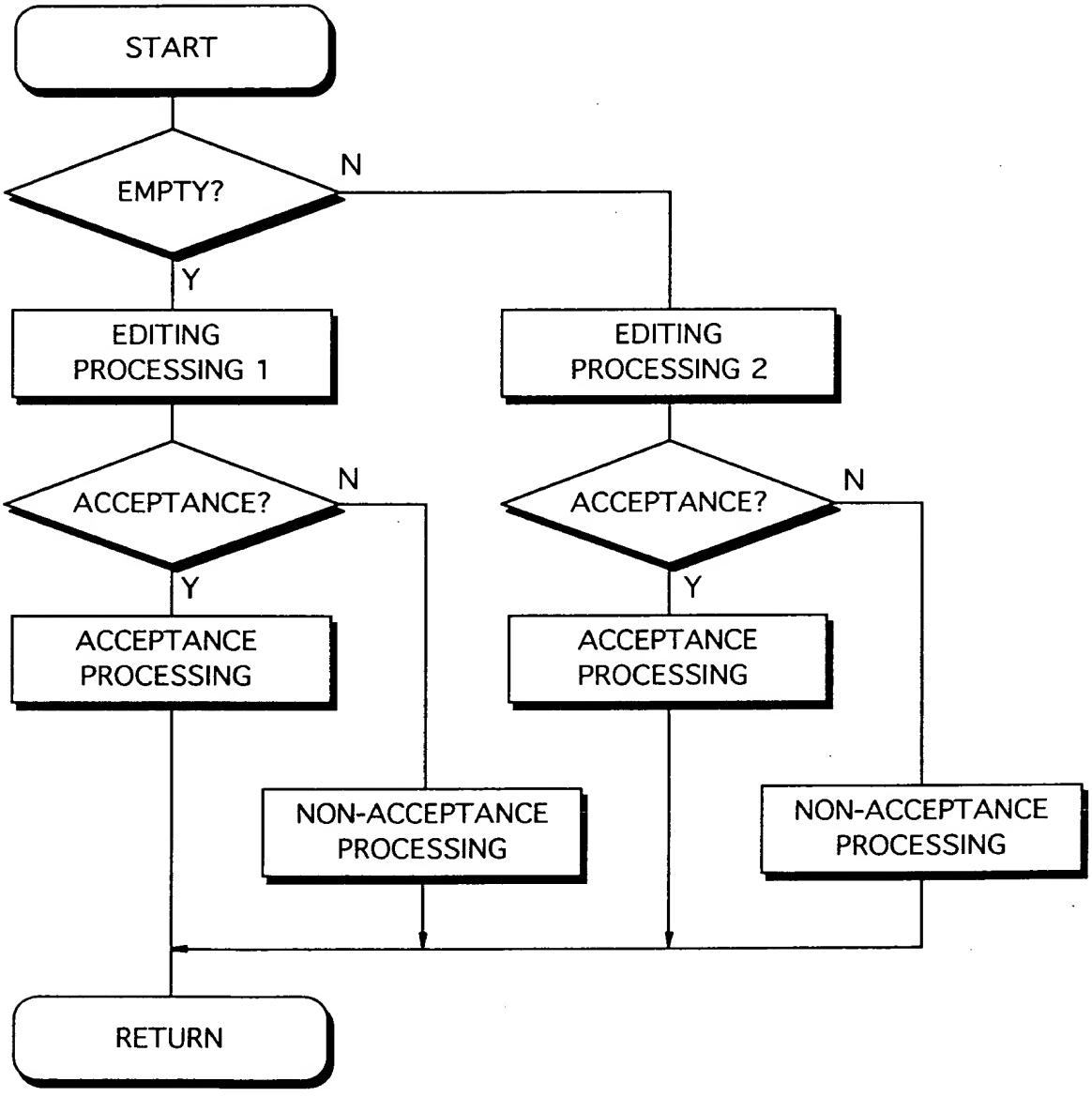


Fig. 13

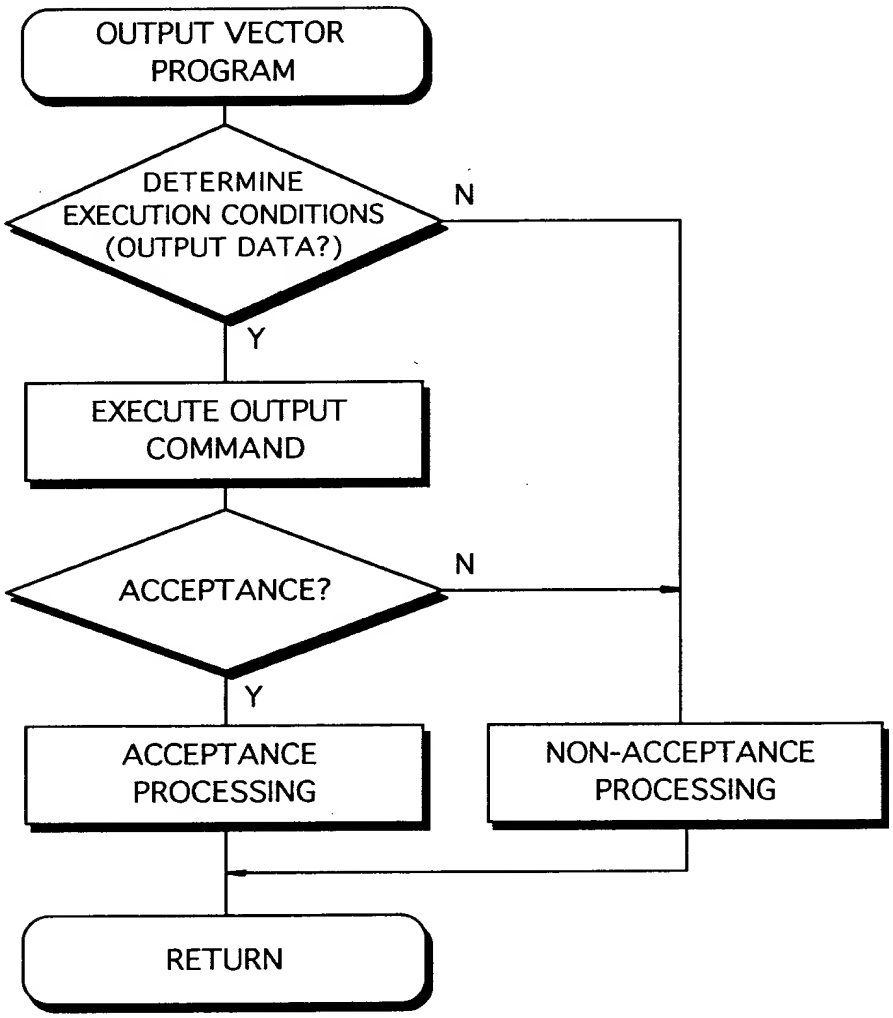

OUTPUT VECTOR PROGRAM
DETERMINE EXECUTION CONDITIONS (OUTPUT DATA?)
N
Y
EXECUTE OUTPUT COMMAND
ACCEPTANCE?
N
Y
ACCEPTANCE PROCESSING
NON-ACCEPTANCE PROCESSING
RETURN

Fig. 14

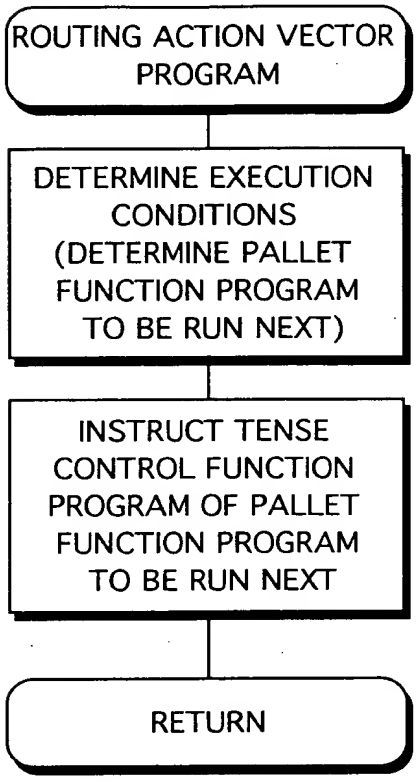

ROUTING ACTION VECTOR PROGRAM
DETERMINE EXECUTION CONDITIONS (DETERMINE PALLET FUNCTION PROGRAM TO BE RUN NEXT)
INSTRUCT TENSE CONTROL FUNCTION PROGRAM OF PALLET FUNCTION PROGRAM TO BE RUN NEXT
RETURN

Fig. 15

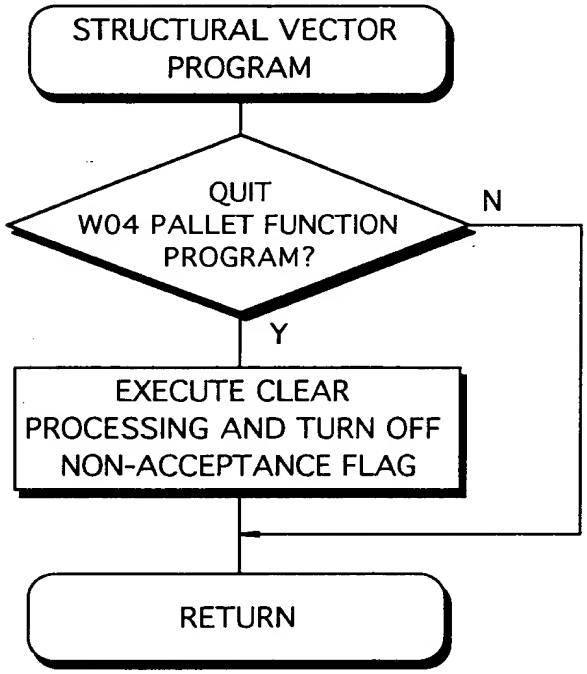

STRUCTURAL VECTOR PROGRAM
QUIT W04 PALLET FUNCTION PROGRAM?
N
Y
EXECUTE CLEAR PROCESSING AND TURN OFF NON-ACCEPTANCE FLAG
RETURN

Fig. 16

HOMOGENEITY VECTOR PROGRAM (W02)

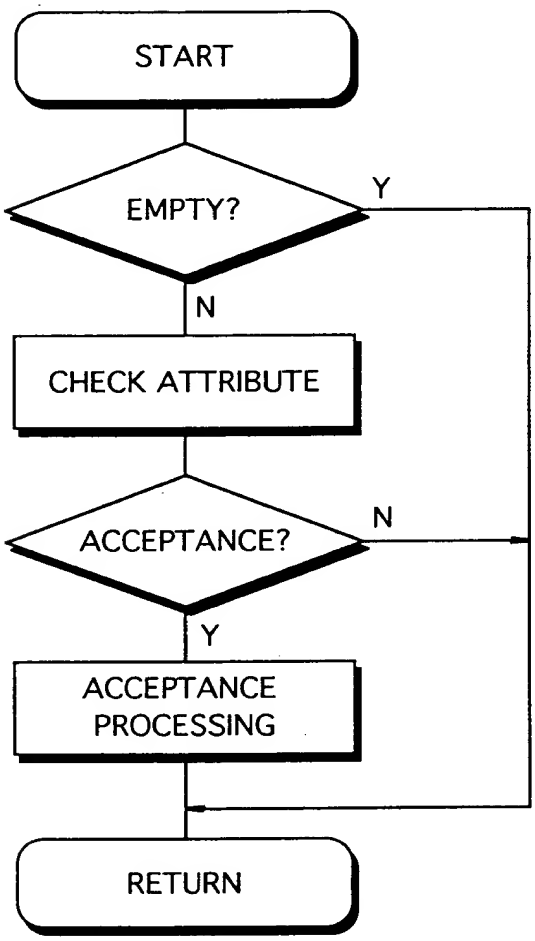


Fig. 17

HOMOGENEITY VECTOR PROGRAM (W02G)

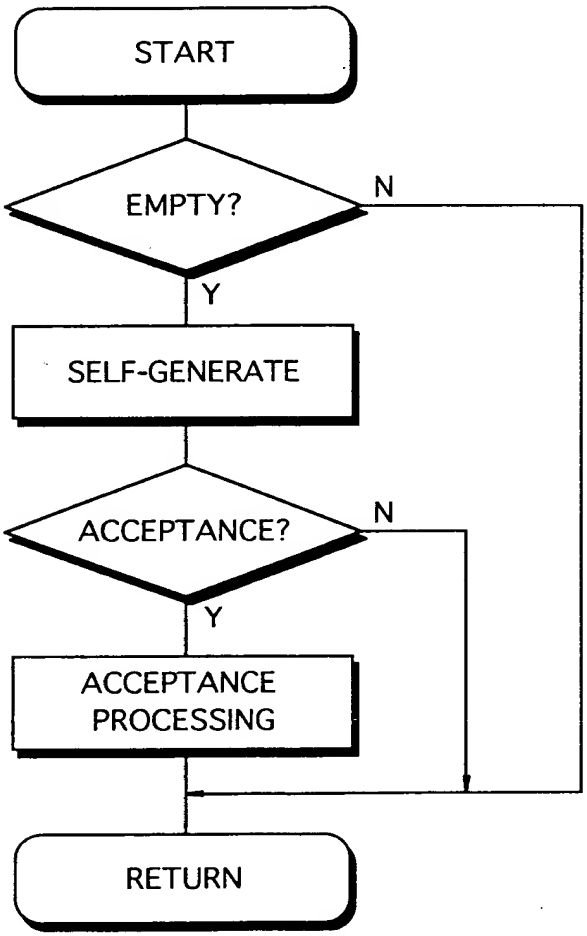


Fig. 18

```
IF W02.□ = LOW-VALUE
 GO TO EXIT.
END-IF
/*CHECK ATTRIBUTE
IF W02.□ = NUMERIC
 GO TO EXIT.
END-IF
IF W02.□ NOT = LOW-VALUE
 GO TO EXIT.
END-IF
W02.□ CNT = W02.□_CNT+1
IF W02.□_CNT <W02_RECALL_MAX
 W02_RECALL_FLG = "1"
ELSE
 W02.□_Non = "1"
END-IF
```

Fig. 19

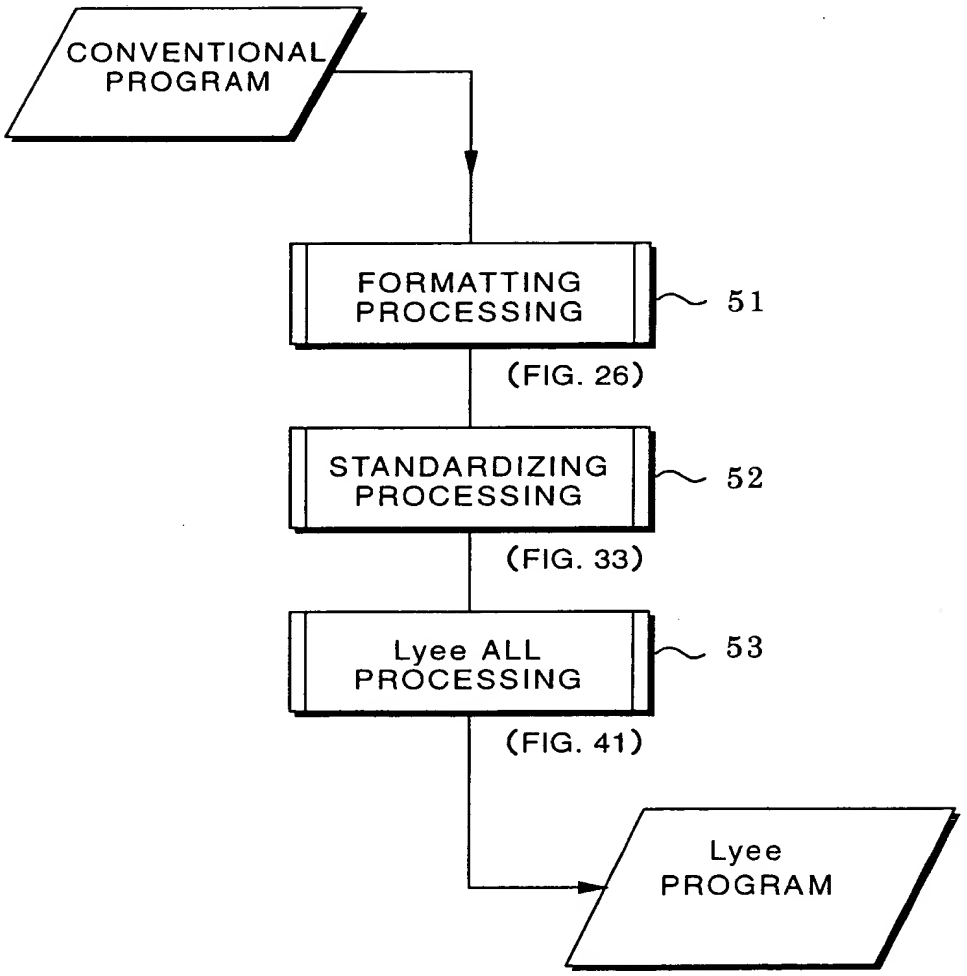

CONVENTIONAL PROGRAM
FORMATTING PROCESSING
51
(FIG. 26)
STANDARDIZING PROCESSING
52
(FIG. 33)
Lyee ALL PROCESSING
53
(FIG. 41)
Lyee PROGRAM

Fig. 20

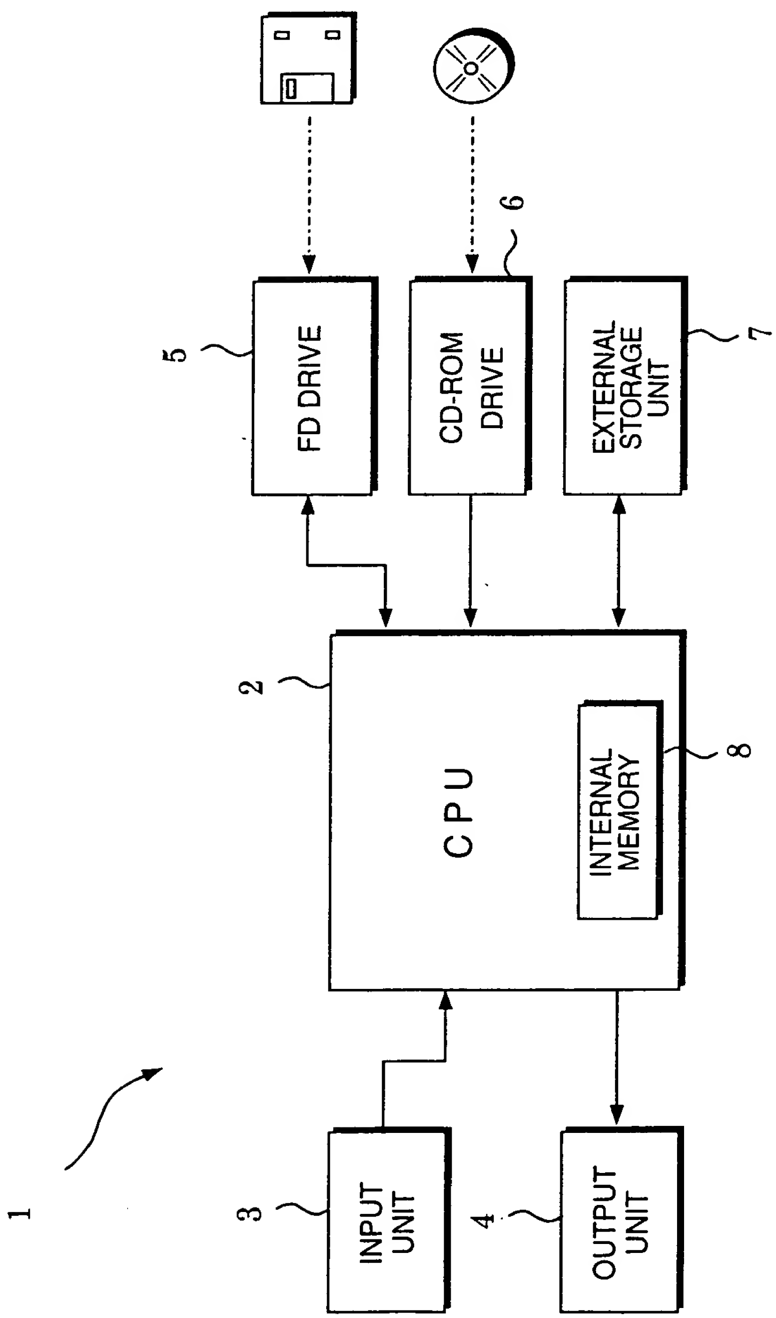

1
FD DRIVE
5
CD-ROM DRIVE
6
EXTERNAL STORAGE UNIT
7
C P U
2
INTERNAL MEMORY
8
INPUT UNIT
3
OUTPUT UNIT
4

Fig. 21

```
01    IDENTIFICATION DIVISION.
02    PROGURAMU-ID. AAA1.
03    ENVIRONMENT DIVISION.
04    CONFIGURATION SECTION.
05       SOURCE-COMPUTER. AS400.
06       OBJECT-COMPUTER. AS400.
07    INPUT-OUTPUT SECTION.
08    FILE-CONTROL.
09       SELECT GAMEN-F ASSIGN TO SCREEN-AAA.
10          ORGANIZATION IS TRANSACTION.

11    DATA DIVISION.
12    FILE SECTION.
13    FD GAMEN-F.
14    01 GAMEN-R.
15       03 SHIN-CD  PIC X(05).
16       03 SU       PIC S9(02).
17       03 TANKA    PIC S9(05).
18       03 KINGAKU  PIC S9(05).

19    WORKING-STRAGE SECTION.
20    01 SHIN-TBL.
21       03 SHIN-CD  PIC X(05).
22       03 TANKA    PIC S9(05).
23    01 END-BTN    PIC X(01).
24    01 WK         PIC S9(05).

01    PROCEDURE DIVISION.
02    MAIN-AA SECTION.
03    MAIN-START.
04       OPEN I-O   GAMEN-F.
05       INITIALIZE GAMEN-R.
06       WRITE GAMEN-R.
07    LOOP-1.
08       READ  GAMEN-F.
09       IF END-BTN = "1"
10          CLOSE GAMEN-F
11          GO TO MAIN-EXIT
12       END-IF.
13       IF SHIN-CD OF GAMEN-R = SPACE OR SU OF GAMEN-R = ZERO
14          MOVE 99999 TO KINGAKU OF GAMEN-R
15       ELSE
16          MOVE SHIN-CD OF GAMEN-R TO SHIN-CD OF SHIN-TBL
17          SELECT TEIKA FROM SHIN-DB INTO :SHIN-TBL.TANKA
18          IF STATUS NOT = ZERO
19             MOVE 99999 TO TANKA OF GAMEN-R
20          ELSE
21             MOVE TANKA OF SHIN-TBL TO TANKA OF GAMEN-R
22             COMPUTE WK = TANKA OF GAMEN-R * SU OF GAMEN-R
23             IF WK > 10000
24                COMPUTE KINGAKU OF GAMEN-R = WK * 0.8
25             ELSE
26                COMPUTE KINGAKU OF GAMEN-R = WK * 0.9
27             END-IF
28          END-IF
29       END-IF.
30       WRITE GAMEN-R.
31       GO TO LOOP-1.
32    MAIN-EXIT.
33    STOP RUN.
```

Fig. 22

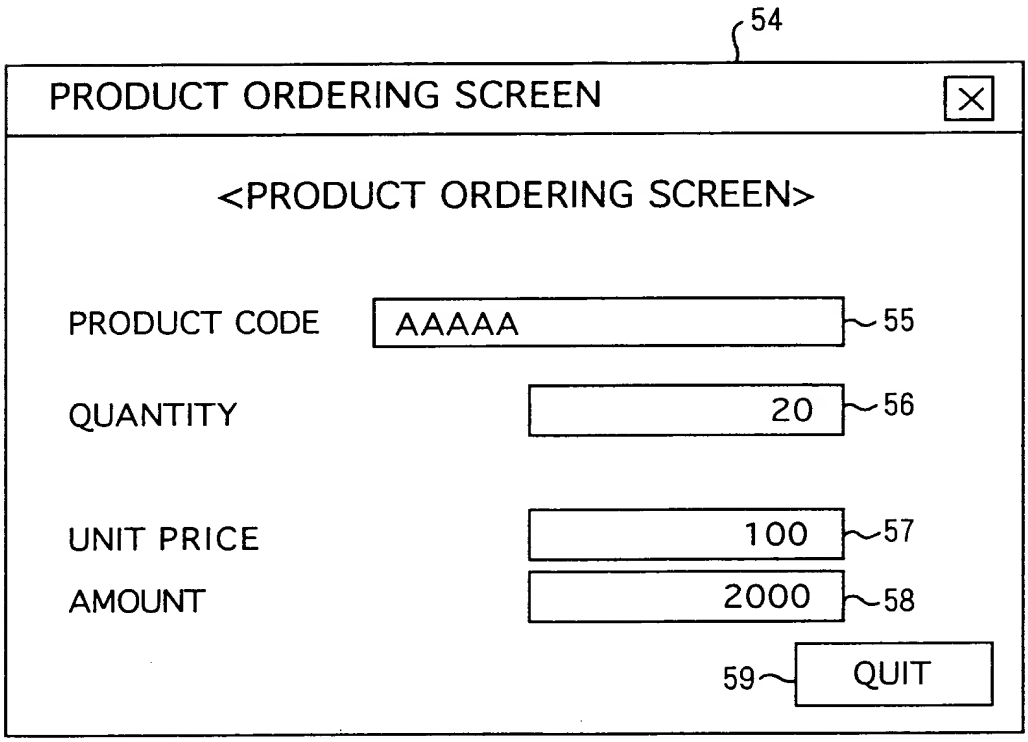

54
PRODUCT ORDERING SCREEN
<PRODUCT ORDERING SCREEN>
PRODUCT CODE
AAAAA
55
QUANTITY
20
56
UNIT PRICE
100
57
AMOUNT
2000
58
59
QUIT

Fig. 23

SHIN-TBL (PRODUCT TABLE)

SHIN-CD (PRODUCT CODE)	TANKA (UNIT PRICE)
AAAAA	100
BBBBB	200
CCCCC	300
DDDDD	400
EEEEE	500
⋮	⋮

60

Fig. 24

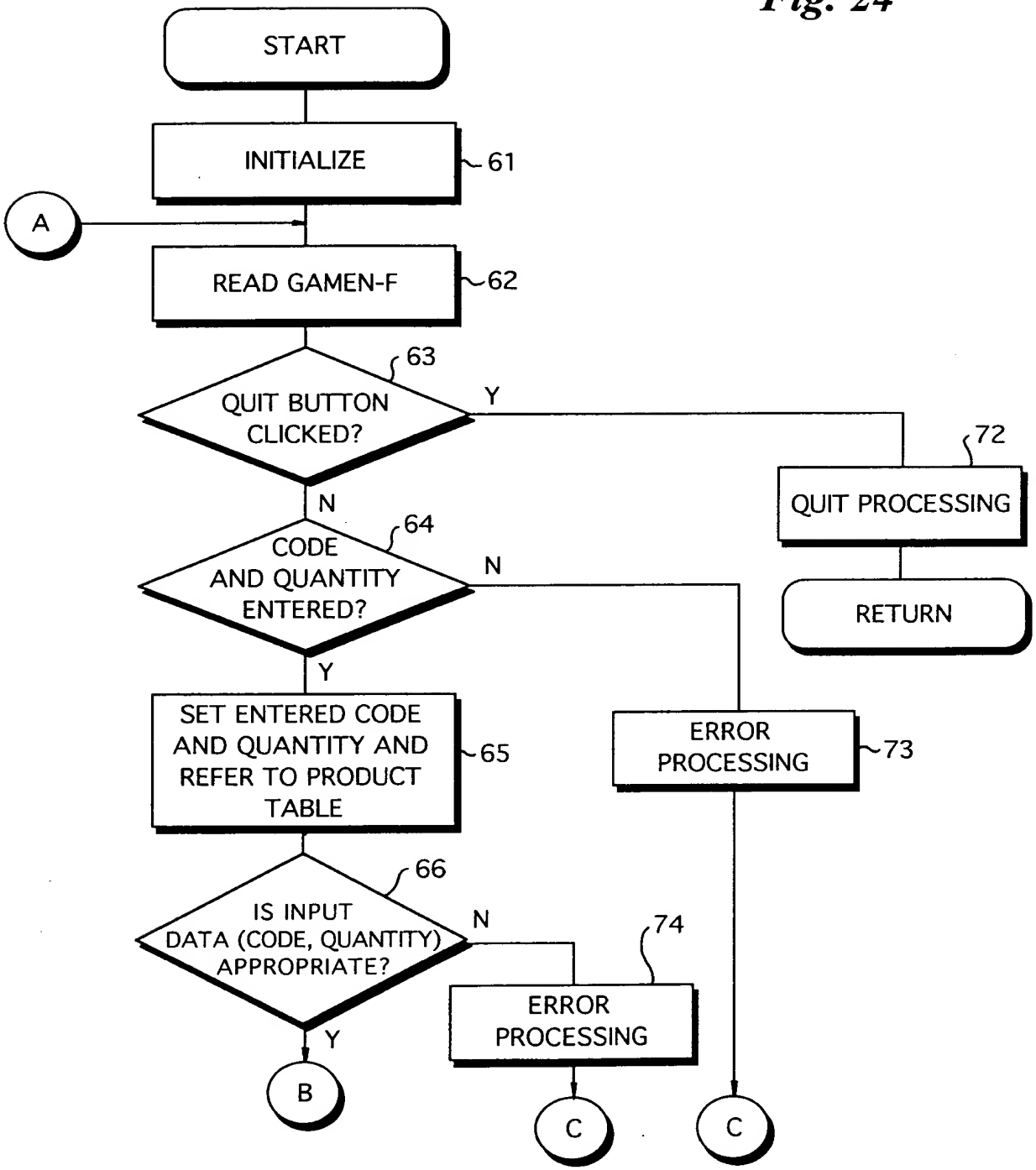

START
INITIALIZE
61
A
READ GAMEN-F
62
63
QUIT BUTTON CLICKED?
Y
72
QUIT PROCESSING
RETURN
N
64
CODE AND QUANTITY ENTERED?
N
Y
SET ENTERED CODE AND QUANTITY AND REFER TO PRODUCT TABLE
65
ERROR PROCESSING
73
66
IS INPUT DATA (CODE, QUANTITY) APPROPRIATE?
N
74
ERROR PROCESSING
Y
B
C
C

Fig. 25

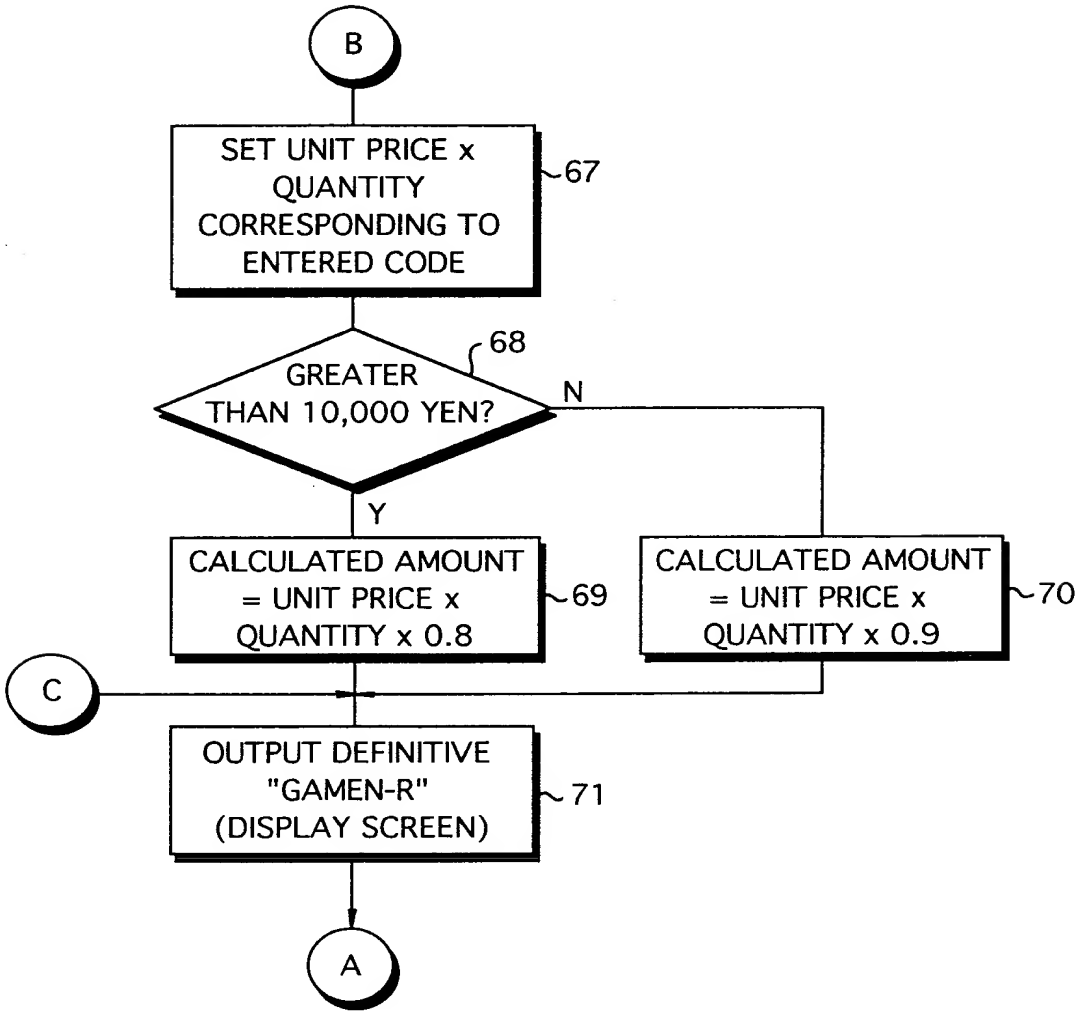

B
SET UNIT PRICE x QUANTITY CORRESPONDING TO ENTERED CODE
67
GREATER THAN 10,000 YEN?
68
N
Y
CALCULATED AMOUNT = UNIT PRICE x QUANTITY x 0.8
69
CALCULATED AMOUNT = UNIT PRICE x QUANTITY x 0.9
70
C
OUTPUT DEFINITIVE "GAMEN-R" (DISPLAY SCREEN)
71
A

Fig. 26

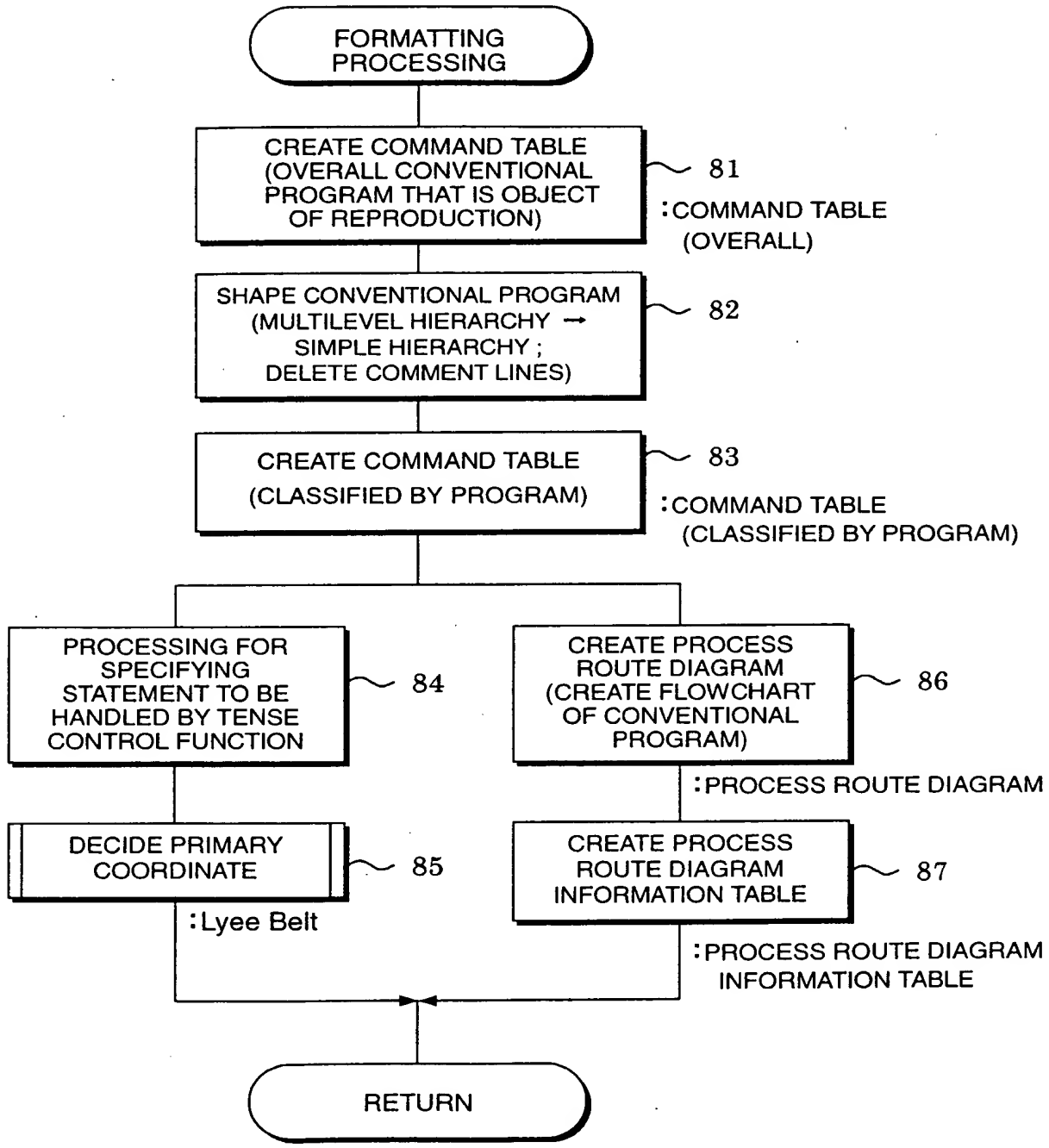

FORMATTING PROCESSING
CREATE COMMAND TABLE (OVERALL CONVENTIONAL PROGRAM THAT IS OBJECT OF REPRODUCTION)
81
:COMMAND TABLE (OVERALL)
SHAPE CONVENTIONAL PROGRAM (MULTILEVEL HIERARCHY → SIMPLE HIERARCHY ; DELETE COMMENT LINES)
82
CREATE COMMAND TABLE (CLASSIFIED BY PROGRAM)
83
:COMMAND TABLE (CLASSIFIED BY PROGRAM)
PROCESSING FOR SPECIFYING STATEMENT TO BE HANDLED BY TENSE CONTROL FUNCTION
84
CREATE PROCESS ROUTE DIAGRAM (CREATE FLOWCHART OF CONVENTIONAL PROGRAM)
86
:PROCESS ROUTE DIAGRAM
DECIDE PRIMARY COORDINATE
85
:Lyee Belt
CREATE PROCESS ROUTE DIAGRAM INFORMATION TABLE
87
:PROCESS ROUTE DIAGRAM INFORMATION TABLE
RETURN

Fig. 27

COMMAND TABLE (OVERALL)

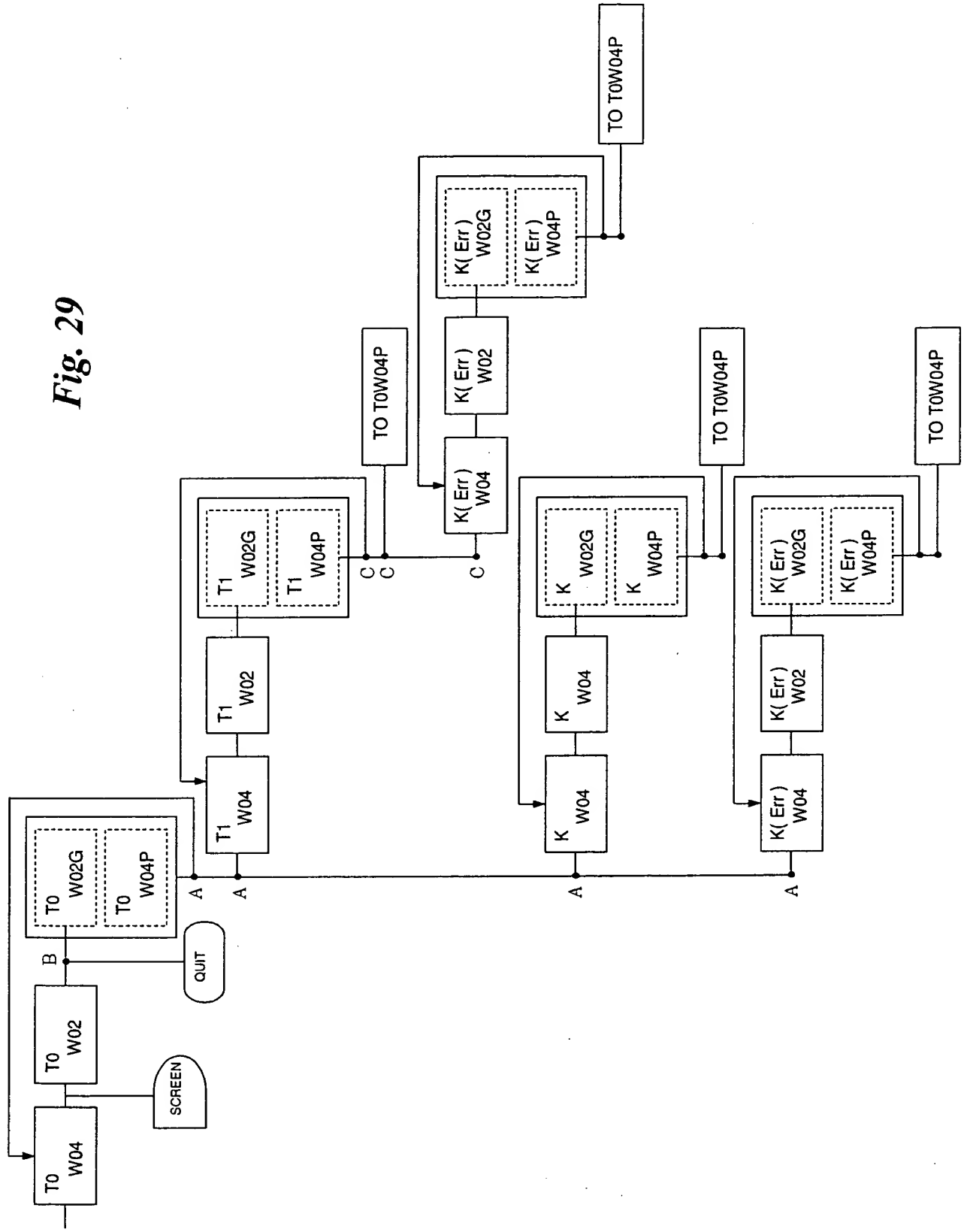
DEFINITIVE TYPE	COMMAND	COMMAND TYPE	CONDITION	INPUT/OUTPUT UNIT	INPUT/OUTPUT	PALLET LOCATION
SCREEN	SELECT	COBOL	ASSIGN TO SCREEN	ASSIGN CLAUSE FILE NAME	INPUT	T0W02
SCREEN	SELECT	COBOL	ASSIGN TO SCREEN	ASSIGN CLAUSE FILE NAME	OUTPUT	T0W04
DB	SELECT	COBOL	FROM	FROM CLAUSE DB NAME	INPUT	T1W02

Fig. 28

COMMAND TABLE (CLASSIFIED BY PROGRAM)

PROGRAM ID	COMMAND	INPUT/OUTPUT UNIT	DEFINITIVE	INPUT/OUTPUT	PALLET LOCATION
AAA1	READ	AAA	GAMEN-R	INPUT	T0W02
AAA1	WRITE	AAA	GAMEN-R	OUTPUT	T0W04
AAA1	SELECT	SHIN-DB	SHIN-TBL	INPUT	T1W02

Fig. 29


TO T0W04P
K(Err) W02G
K(Err) W04P
K(Err) W02
K(Err) W04
TO T0W04P
T1 W02G
T1 W04P
T1 W02
T1 W04
C
C
C
TO T0W04P
K W02G
K W04P
K W04
K W04
TO T0W04P
K(Err) W02G
K(Err) W04P
K(Err) W02
K(Err) W04
A
A
A
A
T0 W02G
T0 W04P
B
QUIT
T0 W02
SCREEN
T0 W04

PROCESS ROUTE DIAGRAM INFORMATION TABLE

Fig. 30

PRESENT PGM ID	PROCESS ROUTE DIAGRAM ID	UNIT-TEAM FUNCTION ID	PALLET ID	ROUTING VECTOR ID	NEXT-PALLET ID	INPUT/OUTPUT COMMAND VECTOR ID	COMMAND	DIFINITIVE
A1	Sample1	Sample_1	A1T0W04	Route-1	A1T0W02			
						WRITE-1	WRITE	GAMEN-R
A1	Sample1	Sample_1	A1T0W02	Route-1	A1T0W03			
				Route-2	STOP			
						READ-1	READ	GAMEN-R
A1	Sample1	Sample_1	A1T0W03	Route-1	A1T0W04			
				Route-2	A1T1W04			
				Route-3	A1K1W04			
				Route-4	A1E1W04			
A1	Sample2	Sample_2	A1T1W04	Route-1	A1T1W02			
A1	Sample2	Sample_2	A1T1W02	Route-1	A1T1W03			
						READ-1	SELECT	SHIN-DB
A1	Sample2	Sample_2	A1T1W03	Route-1	A1T0W04P			
				Route-2	A1E2W04			
A1	Sample3	Sample_3	A1K1W04	Route-1	A1K1W02			
A1	Sample3	Sample_3	A1K1W02	Route-1	A1K1W03			
A1	Sample3	Sample_3	A1K1W03	Route-1	A1T0W04P			
A1	Sample4	Sample_4	A1E1W04	Route-1	A1E1W02			
A1	Sample4	Sample_4	A1E1W02	Route-1	A1E1W03			
A1	Sample4	Sample_4	A1E1W03	Route-1	A1T0W04P			
A1	Sample5	Sample_5	A1E2W04	Route-1	A1E2W02			
A1	Sample5	Sample_5	A1E2W02	Route-1	A1E2W03			
A1	Sample5	Sample_5	A1E2W03	Route-1	A1T0W04P			

Fig. 31

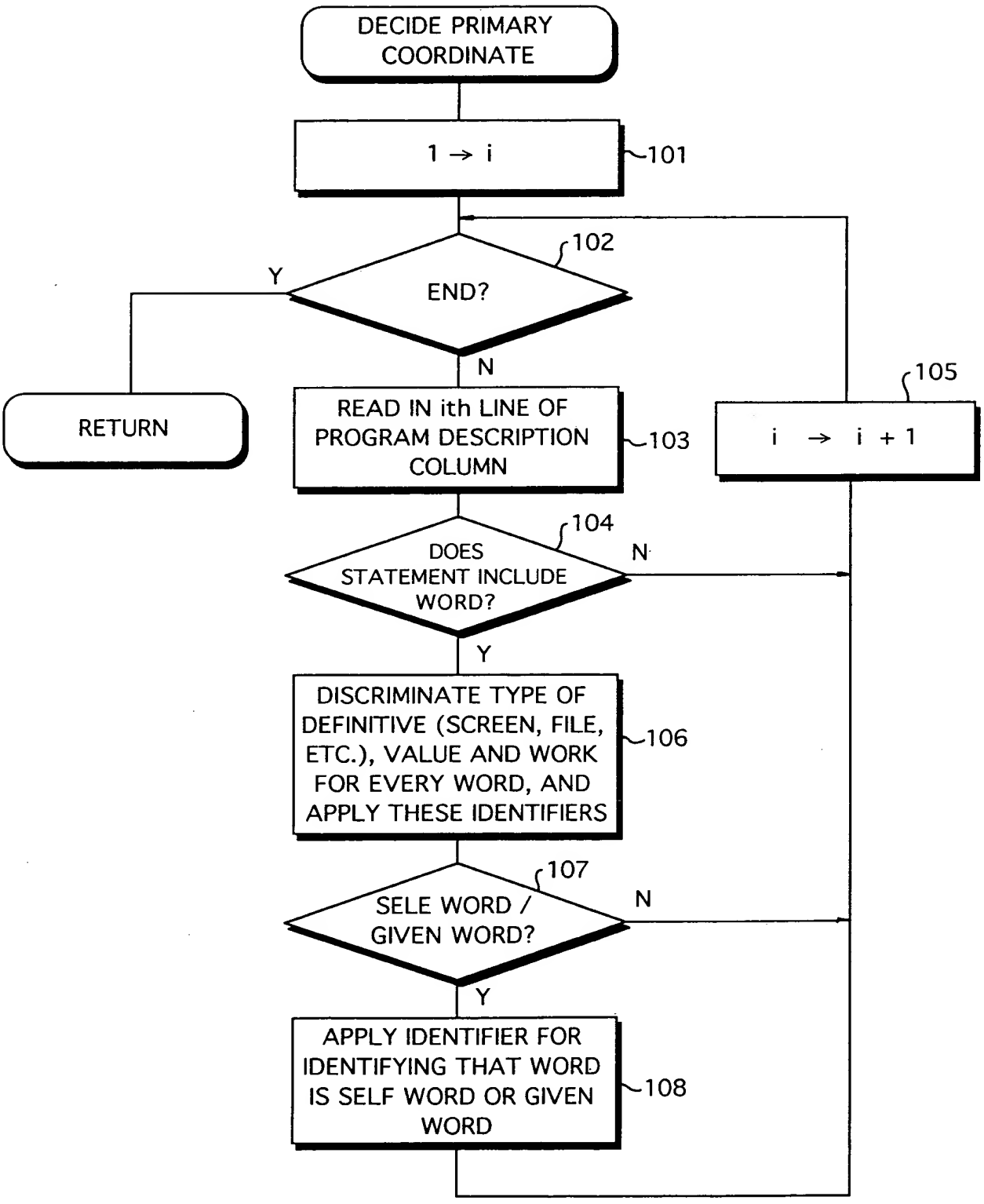

DECIDE PRIMARY COORDINATE
1 → i
101
102
END?
Y
N
RETURN
READ IN ith LINE OF PROGRAM DESCRIPTION COLUMN
103
105
i → i + 1
104
DOES STATEMENT INCLUDE WORD?
N
Y
DISCRIMINATE TYPE OF DEFINITIVE (SCREEN, FILE, ETC.), VALUE AND WORK FOR EVERY WORD, AND APPLY THESE IDENTIFIERS
106
107
SELE WORD / GIVEN WORD?
N
Y
APPLY IDENTIFIER FOR IDENTIFYING THAT WORD IS SELF WORD OR GIVEN WORD
108

Fig. 32

LINE NUMBER	COMMAND TYPE	UNIT-TEAM FUNCTION	PROGRAM DESCRIPTION
01			PROCEDURE DIVISION.
02			MAIN-AA SECTION.
03			MAIN-START.
04		Φ	OPEN I-O GAMEN-F.
05		Φ	INITIALIZE GAMEN-R.
06		Φ	WRITE GAMEN-R.
07			LOOP-1.
08		Φ	READ GAMEN-F.
09	IF		IF END-BTN(SCREEN) = "1"(VALUE)
10		Φ	CLOSE GAMEN-F
11			GO TO MAIN-EXIT
12			END-IF.
13	IF		IF SHIN-CD OF GAMEN-R(SCREEN) = SPACE(VALUE)
			OR SU OF GAMEN-R(SCREEN) = ZERO(VALUE)
14	SELF COMMAND		MOVE 99999(GIVEN : VALUE) TO KINGAKU OF GAMEN-R(SELF : SCREEN)
15			ELSE
16	SELF COMMAND		MOVE SHIN-CD OF GAMEN-R(GIVEN : SCREEN)
			TO SHIN-CD OF SHIN-TBL(SELF : CONTROL BOX)
17	CMD		SELECT TEIKA FROM SHIN-DB INTO :SHIN-TBL.TANKA
18	IF		IF STATUS(CONTROL BOX) NOT = ZERO(VALUE)
19	SELF COMMAND		MOVE 99999(SELF : VALUE) TO TANKA OF GAMEN-R(SELF : SCREEN)
20			ELSE
21	SELF COMMAND		MOVE TANKA OF SHIN-TBL(GIVEN:DB)
			TO TANKA OF GAMEN-R(SELF : SCREEN)
22	SELF COMMAND		COMPUTE WK(SELF : WORK)= TANKA OF GAMEN-R(GIVEN : SCREEN)
			* SU OF GAMEN-R(GIVEN : SCREEN)
23	IF		IF WK(WORK) > 10000(VALUE)
24	SELF COMMAND		COMPUTE KINGAKU OF GAMEN-R(SELF:SCREEN) = WK(GIVEN:WORK) * 0.8
25			ELSE
26			COMPUTE KINGAKU OF GAMEN-R(SELF:SCREEN) = WK(GIVEN:WORK) * 0.9
27			END-IF
28			END-IF
29			END-IF.
30		Φ	WRITE GAMEN-R.
31			GO TO LOOP-1.
32			MAIN-EXIT.
33			STOP RUN.

Fig. 33

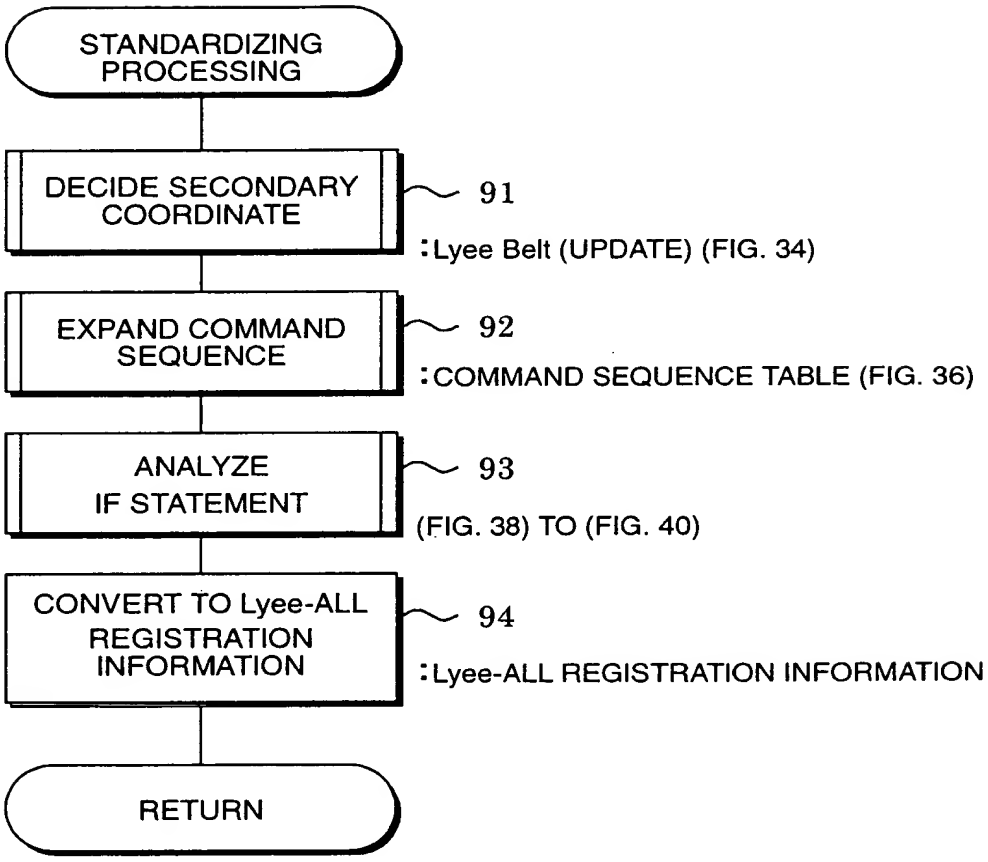

STANDARDIZING PROCESSING
DECIDE SECONDARY COORDINATE
91
:Lyee Belt (UPDATE) (FIG. 34)
EXPAND COMMAND SEQUENCE
92
:COMMAND SEQUENCE TABLE (FIG. 36)
ANALYZE IF STATEMENT
93
(FIG. 38) TO (FIG. 40)
CONVERT TO Lyee-ALL REGISTRATION INFORMATION
94
:Lyee-ALL REGISTRATION INFORMATION
RETURN

Fig. 34

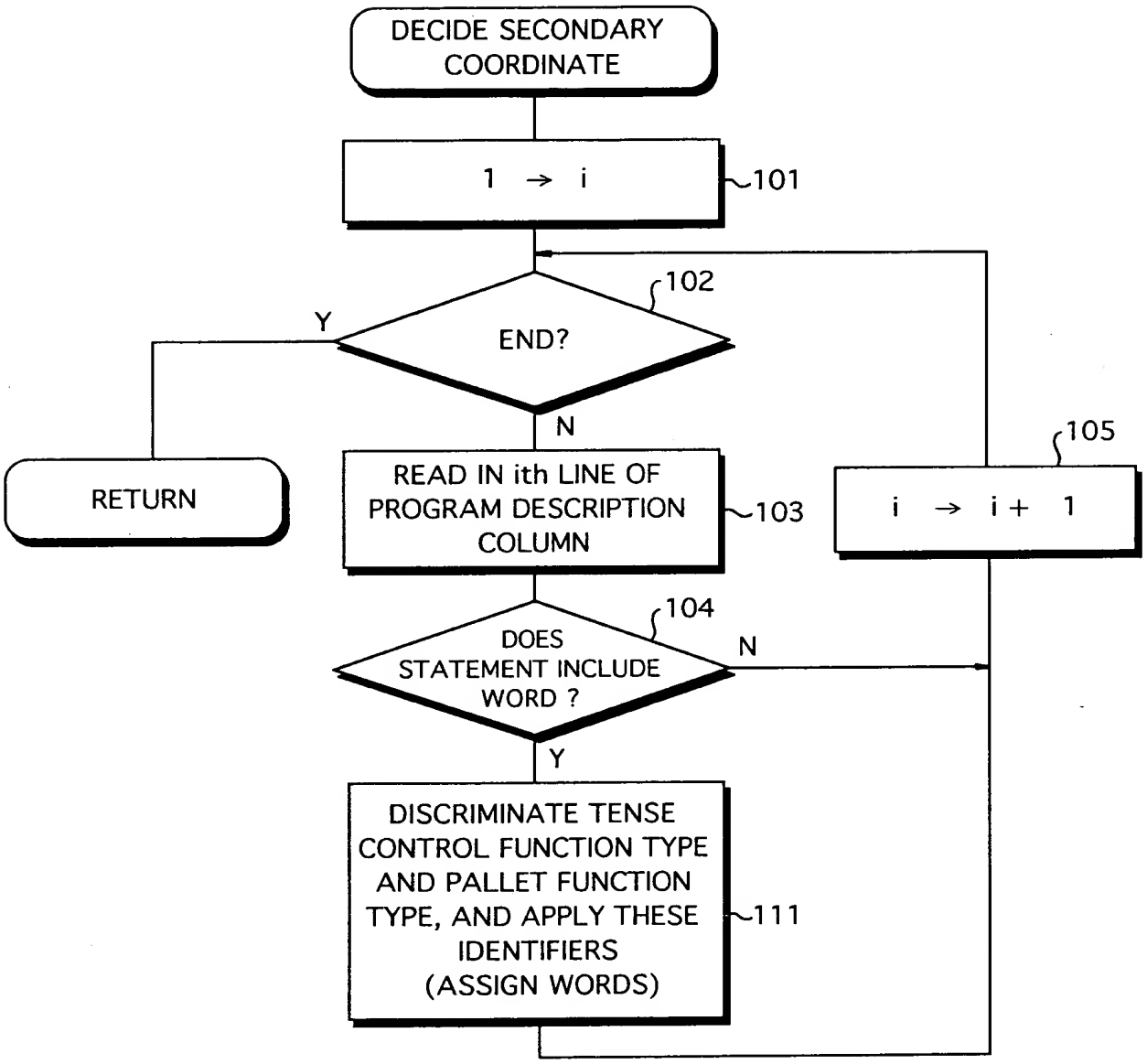

DECIDE SECONDARY COORDINATE
1 → i
101
102
END?
Y
N
RETURN
READ IN ith LINE OF PROGRAM DESCRIPTION COLUMN
103
105
i → i + 1
104
DOES STATEMENT INCLUDE WORD ?
N
Y
DISCRIMINATE TENSE CONTROL FUNCTION TYPE AND PALLET FUNCTION TYPE, AND APPLY THESE IDENTIFIERS (ASSIGN WORDS)
111

Fig. 35

LINE NUMBER	COMMAND TYPE	UNIT-TEAM FUNCTION	PROGRAM DESCRIPTION
01			PROCEDURE DIVISION.
02			MAIN-AA SECTION.
03			MAIN-START.
04		Φ	OPEN I-O GAMEN-F.
05		Φ	INITIALIZE GAMEN-R.
06		Φ	WRITE GAMEN-R.
07			LOOP-1.
08		Φ	READ GAMEN-F.
09	IF		IF END-BTN(T0W02) = "1"(VALUE)
10		Φ	CLOSE GAMEN-F
11			GO TO MAIN-EXIT
12			END-IF.
13	IF		IF SHIN-CD OF GAMEN-R(T0W02) = SPACE(VALUE)
			OR SU OF GAMEN-R(T0W02) = ZERO(VALUE)
14	SELF COMMAND		MOVE 99999(GIVEN : VALUE) TO KINGAKU OF GAMEN-R(SELF:T0W04)
15			ELSE
16	SELF COMMAND		MOVE SHIN-CD OF GAMEN-R(GIVEN:T0W02)
			TO SHIN-CD OF SHIN-TBL.(SELF:T1CB)
17	CMD		SELECT TEIKA FROM SHIN-DB INTO :SHIN-TBL.TANKA
18	IF		IF STATUS(T1CB) NOT = ZERO(VALUE)
19	SELF COMMAND		MOVE 99999(GIVEN:VALUE) TO TANKA OF GAMEN-R(SELF:T0W04)
20			ELSE
21	SELF COMMAND		MOVE TANKA OF SHIN-TBL(GIVEN:T1W02)
			TO TANKA OF GAMEN-R(SELF:T0W04)
22	SELF COMMAND		COMPUTE WK(SELF:KW04P)= TANKA OF GAMEN-R(GIVEN:T0W04)
			* SU OF GAMEN-R(GIVEN:T0W02)
23	IF		IF WK(KW04P) > 10000(VALUE)
24	SELF COMMAND		COMPUTE KINGAKU OF GAMEN-R(SELF:T0W04) = WK(GIVEN:KW04P) * 0.8
25			ELSE
26			COMPUTE KINGAKU OF GAMEN-R(SELF:T0W04) = WK(GIVEN:KW04P) * 0.9
27			END-IF
28			END-IF
29			END-IF.
30		Φ	WRITE GAMEN-R.
31			GO TO LOOP-1.
32			MAIN-EXIT.
33			STOP RUN.

Fig. 36

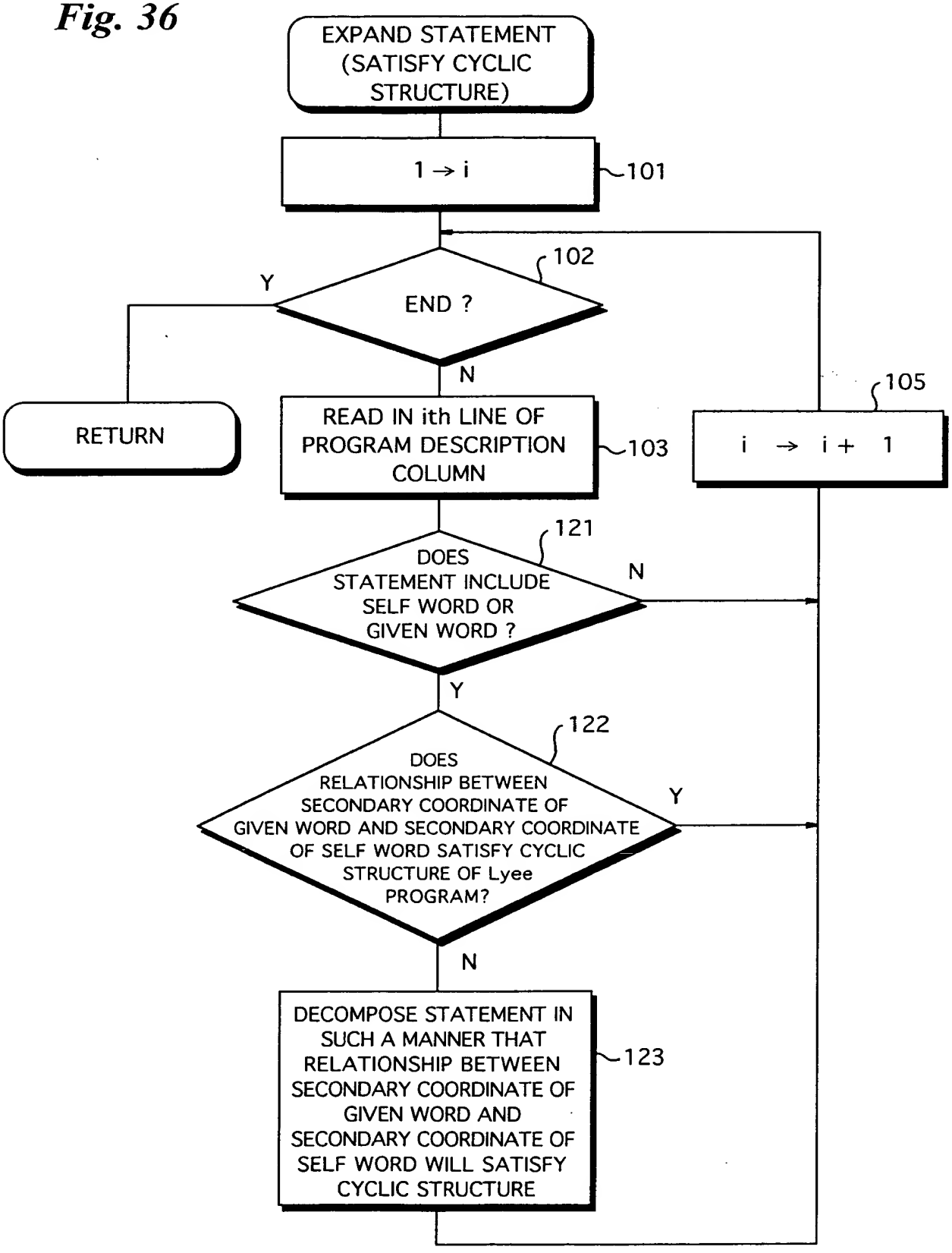

EXPAND STATEMENT (SATISFY CYCLIC STRUCTURE)
1 → i
101
102
END ?
Y
N
RETURN
READ IN ith LINE OF PROGRAM DESCRIPTION COLUMN
103
105
i → i + 1
121
DOES STATEMENT INCLUDE SELF WORD OR GIVEN WORD ?
N
Y
122
DOES RELATIONSHIP BETWEEN SECONDARY COORDINATE OF GIVEN WORD AND SECONDARY COORDINATE OF SELF WORD SATISFY CYCLIC STRUCTURE OF Lyee PROGRAM?
Y
N
DECOMPOSE STATEMENT IN SUCH A MANNER THAT RELATIONSHIP BETWEEN SECONDARY COORDINATE OF GIVEN WORD AND SECONDARY COORDINATE OF SELF WORD WILL SATISFY CYCLIC STRUCTURE
123

Fig. 37

COMMAND SEQUENCE TABLE

LINE NUMBER	TYPE	LOCATION	STATEMENT AFTER EXPANSION
21	DUPLICATE VECTOR	T1W02G	MOVE TANKA OF SHIN-TBL(T1W02) TO TANKA OF SHIN-TBL(T1W02G)
21	DUPLICATE VECTOR-TYPE HOMOGENEITY VECTOR	T1W04P	MOVE TANKA OF SHIN-TBL(T1W02G) TO TANKA OF SHIN-TBL(T1W04P BOUNDARY)
21	R-TYPE DUPLICATE VECTOR	T0W04P	MOVE TANKA OF SHIN-TBL(T1W04P BOUNDARY) TO TANKA OF SHIN-TBL(T0W04P BOUNDARY)
21	HOMOGENEITY VECTOR	T0W04P	MOVE TANKA OF SHIN-TBL(T0W04P BOUNDARY) MOVE TANKA TO TANKA OF GAMEN-R(T0W04P)
21	DUPLICATE VECTOR	T0W04	MOVE TANKA OF GAMEN-R(T0W04P) TO TANKA OF GAMEN-R(T0W04)

Fig. 38

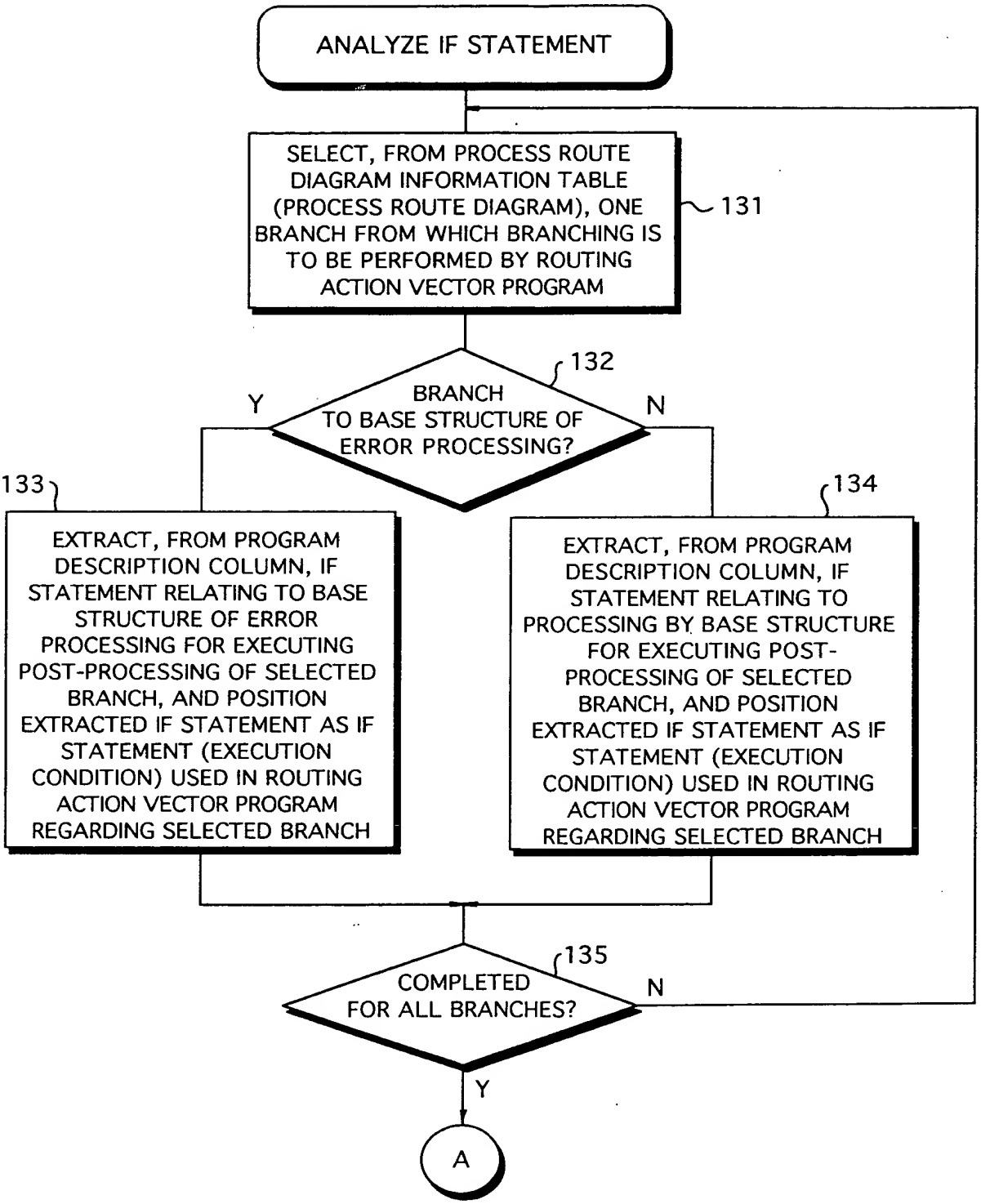

ANALYZE IF STATEMENT
SELECT, FROM PROCESS ROUTE DIAGRAM INFORMATION TABLE (PROCESS ROUTE DIAGRAM), ONE BRANCH FROM WHICH BRANCHING IS TO BE PERFORMED BY ROUTING ACTION VECTOR PROGRAM
131
132
BRANCH TO BASE STRUCTURE OF ERROR PROCESSING?
Y
N
133
EXTRACT, FROM PROGRAM DESCRIPTION COLUMN, IF STATEMENT RELATING TO BASE STRUCTURE OF ERROR PROCESSING FOR EXECUTING POST-PROCESSING OF SELECTED BRANCH, AND POSITION EXTRACTED IF STATEMENT AS IF STATEMENT (EXECUTION CONDITION) USED IN ROUTING ACTION VECTOR PROGRAM REGARDING SELECTED BRANCH
134
EXTRACT, FROM PROGRAM DESCRIPTION COLUMN, IF STATEMENT RELATING TO PROCESSING BY BASE STRUCTURE FOR EXECUTING POST-PROCESSING OF SELECTED BRANCH, AND POSITION EXTRACTED IF STATEMENT AS IF STATEMENT (EXECUTION CONDITION) USED IN ROUTING ACTION VECTOR PROGRAM REGARDING SELECTED BRANCH
135
COMPLETED FOR ALL BRANCHES?
N
Y
A

Fig. 39

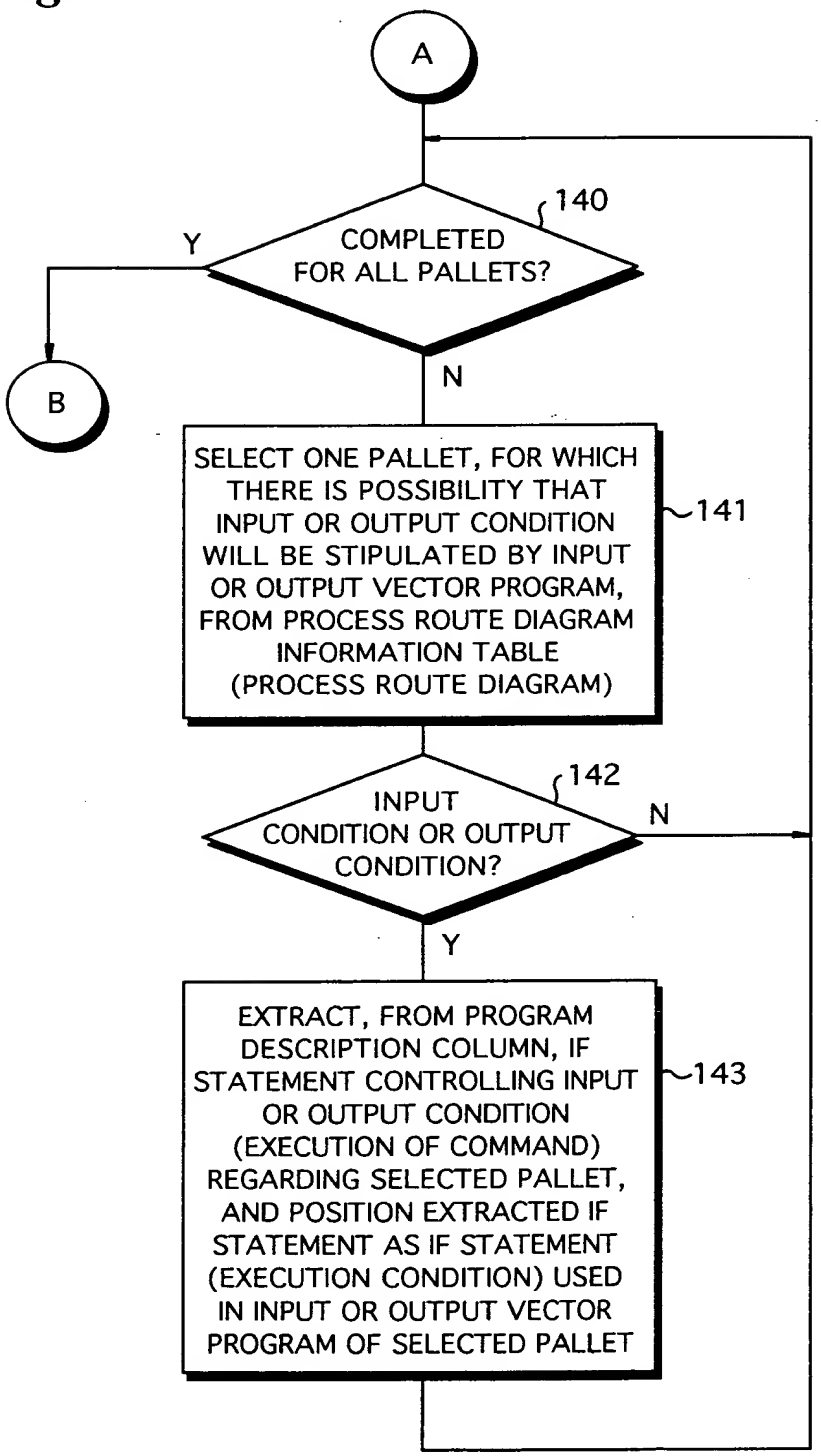

A
140
COMPLETED FOR ALL PALLETS?
Y
N
B
SELECT ONE PALLET, FOR WHICH THERE IS POSSIBILITY THAT INPUT OR OUTPUT CONDITION WILL BE STIPULATED BY INPUT OR OUTPUT VECTOR PROGRAM, FROM PROCESS ROUTE DIAGRAM INFORMATION TABLE (PROCESS ROUTE DIAGRAM)
141
142
INPUT CONDITION OR OUTPUT CONDITION?
N
Y
EXTRACT, FROM PROGRAM DESCRIPTION COLUMN, IF STATEMENT CONTROLLING INPUT OR OUTPUT CONDITION (EXECUTION OF COMMAND) REGARDING SELECTED PALLET, AND POSITION EXTRACTED IF STATEMENT AS IF STATEMENT (EXECUTION CONDITION) USED IN INPUT OR OUTPUT VECTOR PROGRAM OF SELECTED PALLET
143

Fig. 40

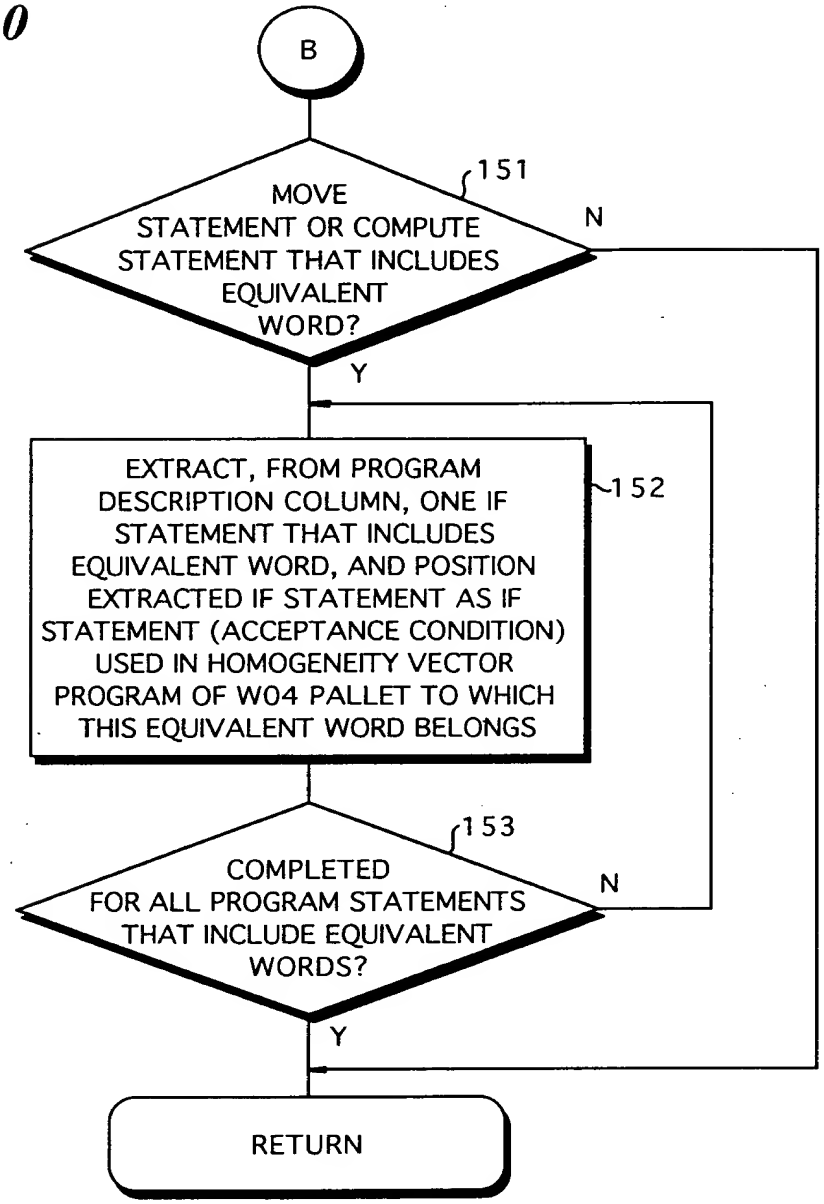

B
151
MOVE STATEMENT OR COMPUTE STATEMENT THAT INCLUDES EQUIVALENT WORD?
N
Y
EXTRACT, FROM PROGRAM DESCRIPTION COLUMN, ONE IF STATEMENT THAT INCLUDES EQUIVALENT WORD, AND POSITION EXTRACTED IF STATEMENT AS IF STATEMENT (ACCEPTANCE CONDITION) USED IN HOMOGENEITY VECTOR PROGRAM OF W04 PALLET TO WHICH THIS EQUIVALENT WORD BELONGS
152
153
COMPLETED FOR ALL PROGRAM STATEMENTS THAT INCLUDE EQUIVALENT WORDS?
N
Y
RETURN

Fig. 41

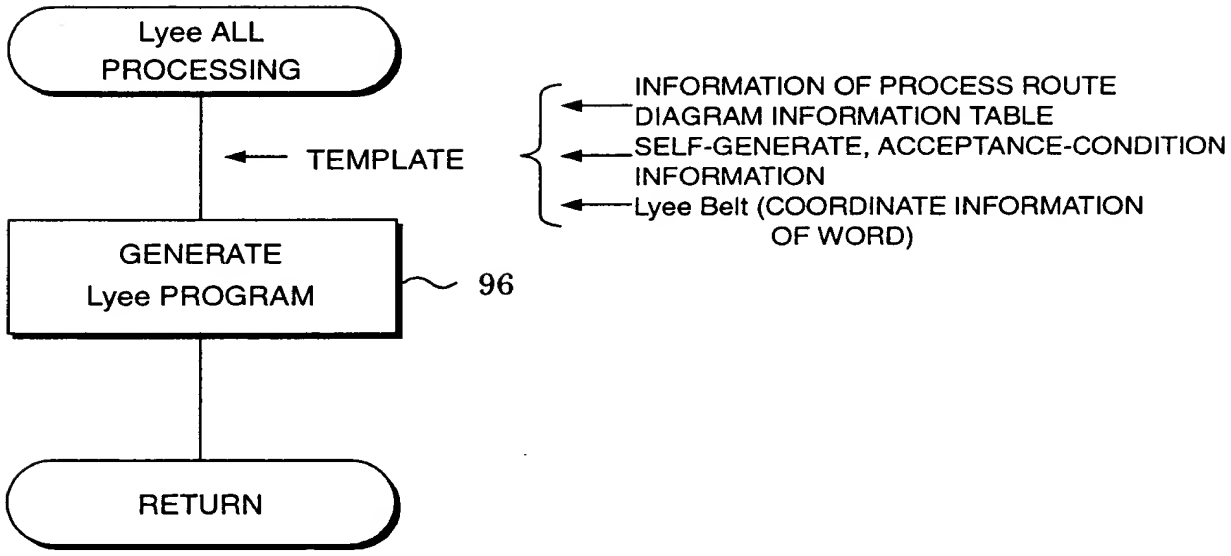

Lyee ALL PROCESSING
TEMPLATE
INFORMATION OF PROCESS ROUTE DIAGRAM INFORMATION TABLE
SELF-GENERATE, ACCEPTANCE-CONDITION INFORMATION
Lyee Belt (COORDINATE INFORMATION OF WORD)
GENERATE Lyee PROGRAM
96
RETURN

Fig. 42

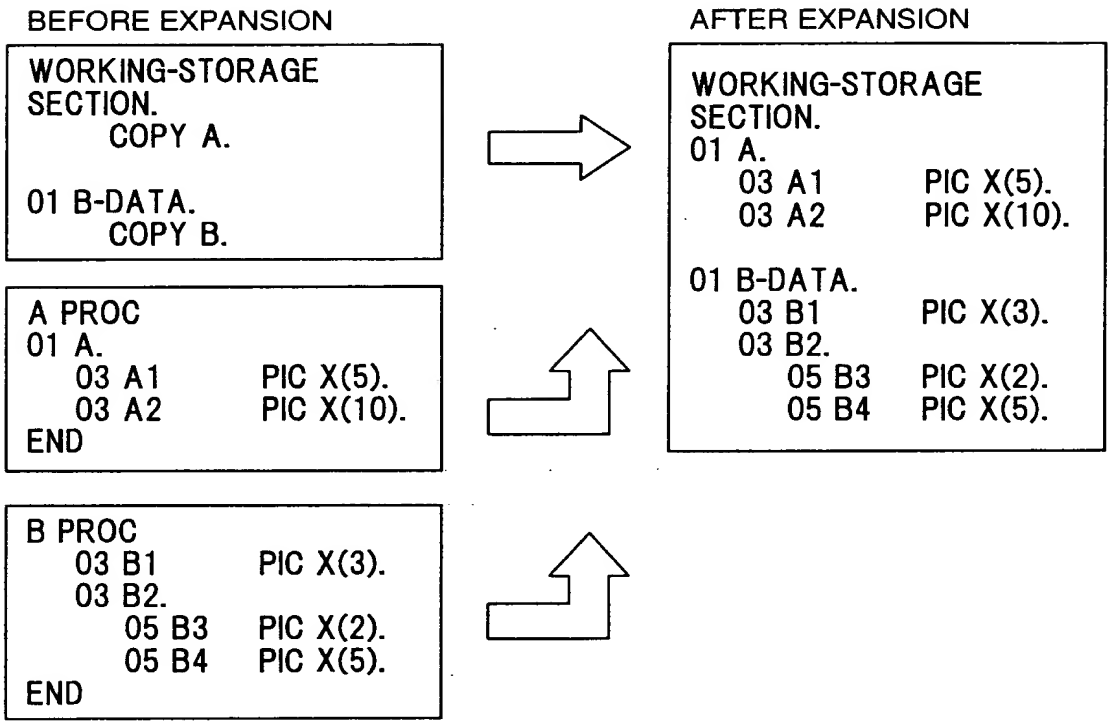

BEFORE EXPANSION
WORKING-STORAGE SECTION.
COPY A.
01 B-DATA.
COPY B.
A PROC
01 A.
03 A1 PIC X(5).
03 A2 PIC X(10).
END
B PROC
03 B1 PIC X(3).
03 B2.
05 B3 PIC X(2).
05 B4 PIC X(5).
END
AFTER EXPANSION
WORKING-STORAGE SECTION.
01 A.
03 A1 PIC X(5).
03 A2 PIC X(10).
01 B-DATA.
03 B1 PIC X(3).
03 B2.
05 B3 PIC X(2).
05 B4 PIC X(5).

Fig. 43

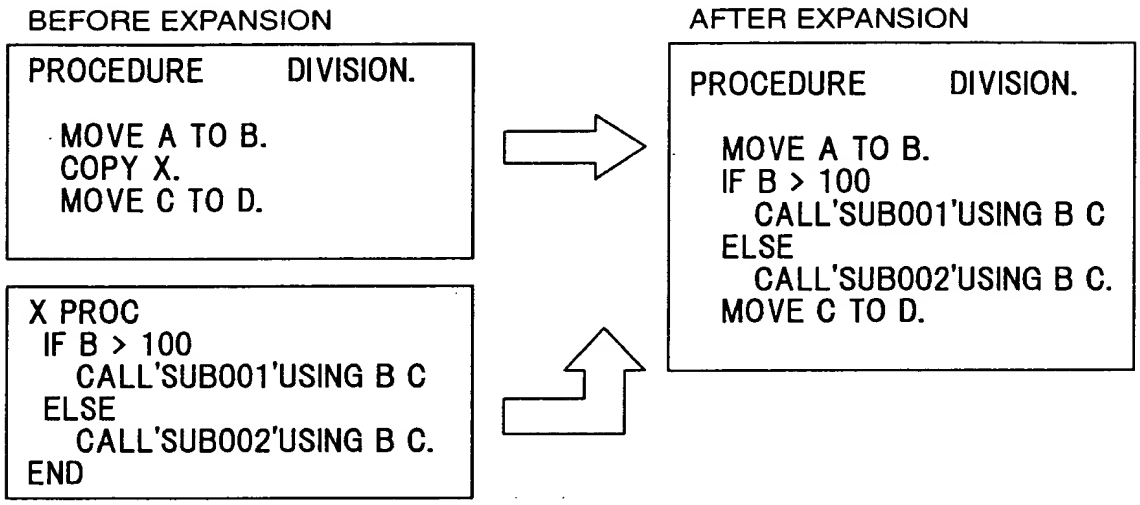

BEFORE EXPANSION
PROCEDURE DIVISION.
MOVE A TO B.
COPY X.
MOVE C TO D.
X PROC
IF B > 100
CALL'SUB001'USING B C
ELSE
CALL'SUB002'USING B C.
END
AFTER EXPANSION
PROCEDURE DIVISION.
MOVE A TO B.
IF B > 100
CALL'SUB001'USING B C
ELSE
CALL'SUB002'USING B C.
MOVE C TO D.

Fig. 44

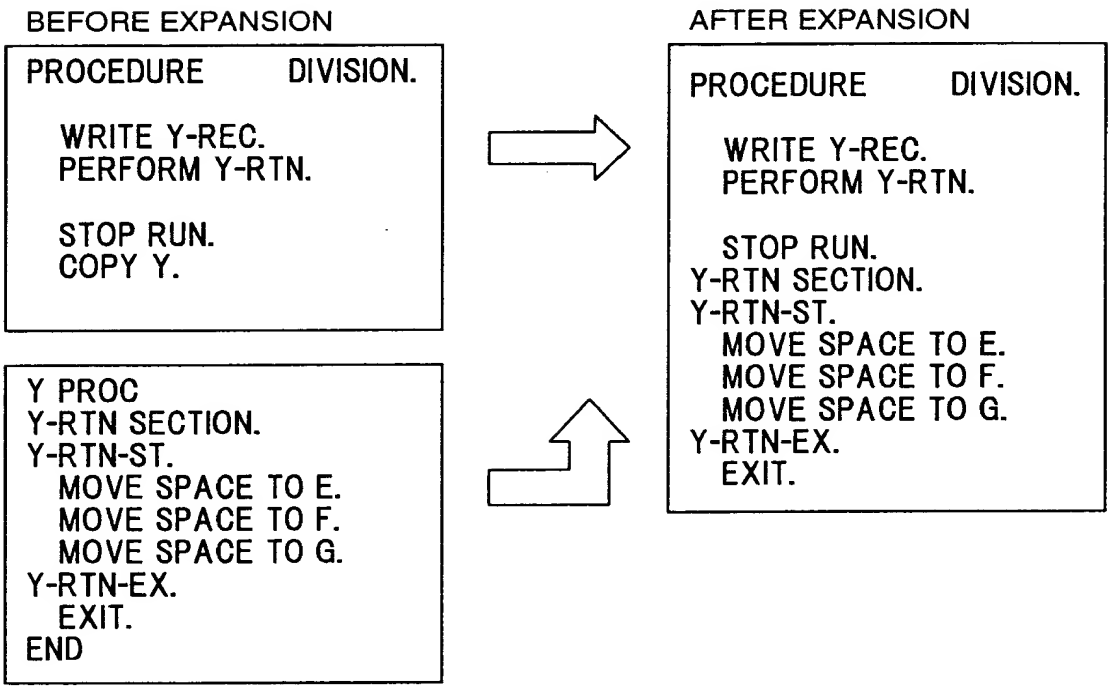

BEFORE EXPANSION
PROCEDURE DIVISION.
WRITE Y-REC.
PERFORM Y-RTN.
STOP RUN.
COPY Y.
Y PROC
Y-RTN SECTION.
Y-RTN-ST.
MOVE SPACE TO E.
MOVE SPACE TO F.
MOVE SPACE TO G.
Y-RTN-EX.
EXIT.
END
AFTER EXPANSION
PROCEDURE DIVISION.
WRITE Y-REC.
PERFORM Y-RTN.
STOP RUN.
Y-RTN SECTION.
Y-RTN-ST.
MOVE SPACE TO E.
MOVE SPACE TO F.
MOVE SPACE TO G.
Y-RTN-EX.
EXIT.

Fig. 45

BEFORE EXPANSION

MAIN PROGRAM

```
WORKING-STORAGE SECTION.
01 WK-A          PIC 9(10).
01 WK-B          PIC 9(10).
01 WK-C          PIC 9(10).
  ....
PROCEDURE       DIVISION.
  ⋮
MOVE A TO WK-A.
MOVE B TO WK-B.
CALL'SUB001'
      USING WK-A WK-B WK-C.
MOVE WK-C TO C.
  ⋮
```

SUBROUTINE

```
PROGRAM-ID.       SUB001.
  ....
LINKAGE SECTION.
01 WK-X          PIC 9(10).
01 WK-Y          PIC 9(10).
01 WK-Z          PIC 9(10).
PROCEDURE       DIVISION.
   USING WK-X WK-Y WK-Z.
PROC-RTN.       SECTION.
PROC-ST.

 COMPUTE
   WK-Z=2 * WK-X * WK-Y *
WK-Y
PROC-EX.
   EXIT PROGRAM.
```

AFTER EXPANSION

MAIN PROGRAM

```
WORKING-STORAGE SECTION.
01 WK-A          PIC 9(10).
01 WK-B          PIC 9(10).
01 WK-C          PIC 9(10).
  ....
PROCEDURE       DIVISION.
  ⋮
MOVE A TO WK-A.
MOVE B TO WK-B.
WK-C=2 * WK-A * WK-B *
WK-B

MOVE WK-C TO C.
  ⋮
```

09/868893

Fig. 46

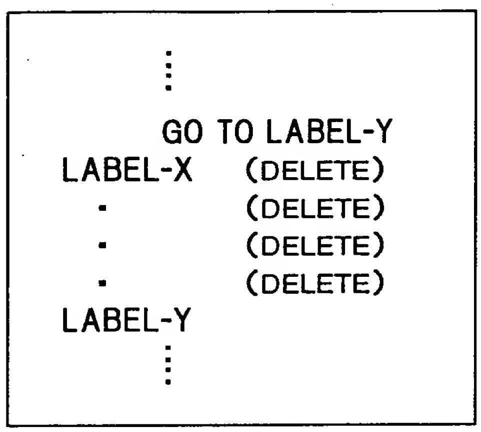


Fig. 47

SET IDX TO 1 ⇒ MOVE 1 TO IDX

Fig. 48

ORIGINAL COMMAND	AFTER SHAPING
MOVE A TO B,C ⇒	MOVE A TO B MOVE A TO C

Fig. 49

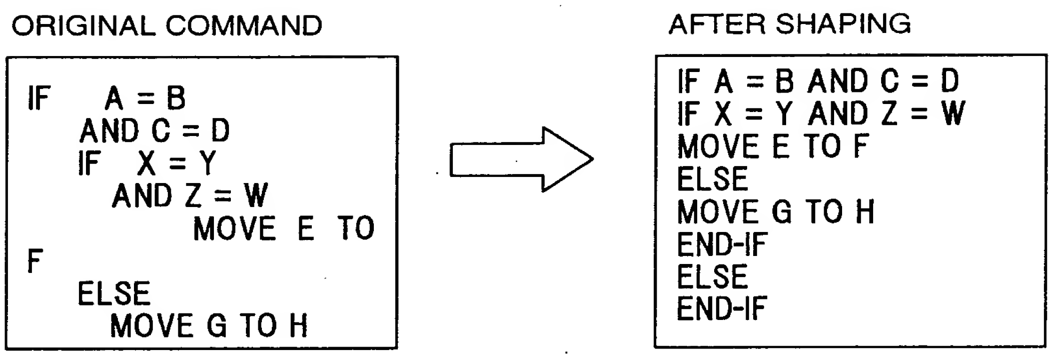
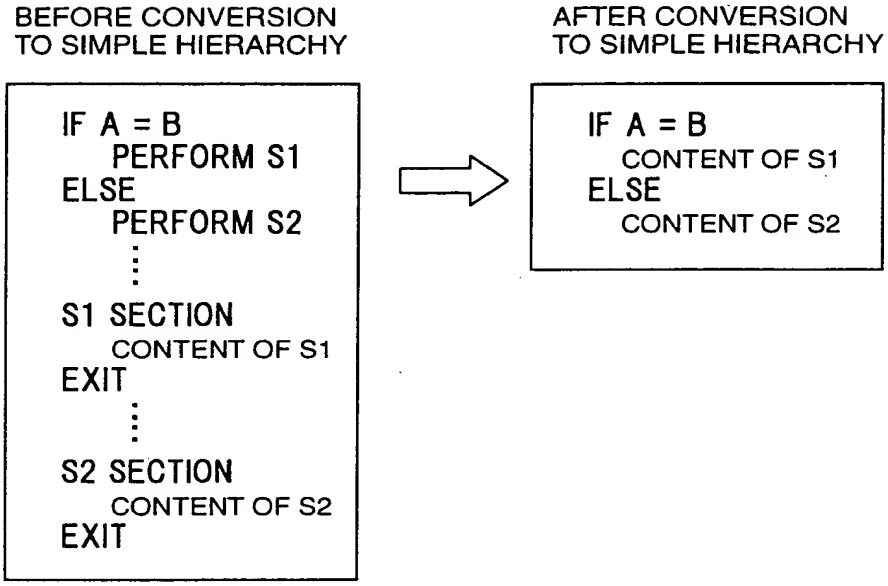

ORIGINAL COMMAND
IF A = B
AND C = D
IF X = Y
AND Z = W
MOVE E TO
F
ELSE
MOVE G TO H
AFTER SHAPING
IF A = B AND C = D
IF X = Y AND Z = W
MOVE E TO F
ELSE
MOVE G TO H
END-IF
ELSE
END-IF

Fig. 50


BEFORE CONVERSION
TO SIMPLE HIERARCHY
IF A = B
PERFORM S1
ELSE
PERFORM S2
S1 SECTION
CONTENT OF S1
EXIT
S2 SECTION
CONTENT OF S2
EXIT
AFTER CONVERSION
TO SIMPLE HIERARCHY
IF A = B
CONTENT OF S1
ELSE
CONTENT OF S2